Information Retrieval, 8, 449–480, 2005

Comparing Rank and Score Combination Methods for Data Fusion in Information Retrieval*

D. FRANK HSU[†‡] hsu@trill.cis.fordham.edu
Department of Computer and Information Science, 113 West 60th Street, LL 813, Fordham University, New York, NY 10023, USA

ISAK TAKSA[§] Isak_Taksa@baruch.cuny.edu
Department of Statistics and Computer Information Systems, Baruch College, One Bernard Baruch Way, Box 11-220, New York, NY 10010, USA

Abstract. Combination of multiple evidences (multiple query formulations, multiple retrieval schemes or systems) has been shown (mostly experimentally) to be effective in data fusion in information retrieval. However, the question of why and how combination should be done still remains largely unanswered. In this paper, we provide a model for simulation and a framework for analysis in the study of data fusion in the information retrieval domain. A *rank/score function* is defined and the concept of a Cayley graph is used in the design and analysis of our framework. The model and framework have led us to better understanding of the data fusion phenomena in information retrieval. In particular, by exploiting the graphical properties of the rank/score function, we have shown analytically and by simulation that combination using rank performs better than combination using score under certain conditions. Moreover, we demonstrated that the rank/score function might be used as a predictive variable for the effectiveness of combination of multiple evidences.

Keywords: information retrieval (IR), data fusion (DF), rank combination, score combination, multiple evidences, evidence combinations, permutation, symmetric group, Cayley graphs and digraphs, rank/score function

1. Introduction

Information retrieval can be considered as a problem of inference (van Rijsbergen 1986). It is a process concerned with estimating, given available evidence about things, such as information need and documents, the likelihood (or probability) of relevance of a document to the information need. As such, different query formulations constitute different sources of evidence that could be used to infer the probable relevance of a document to an information need. This can be generalized to include any source of evidence that might be used for IR such as the evidence of different retrieval techniques, different document representation techniques, or different IR systems.

*Authors wish to dedicate this paper to the memory of our friend and colleague Professor Jacob Shapiro, who passed away September 2003.

[†]*Previous address*: DIMACS Center, Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08854-8018, USA.

[‡]Supported in part by the DIMACS NSF grant STC-91-19999 and by NJ Commission.
[§]Supported in part by a grant from The City University of New York PSC-CUNY Research Award.

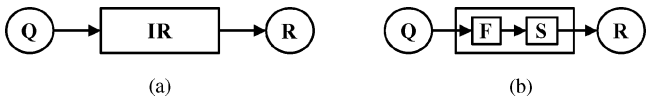


Figure 1. Information retrieval (IR) process.

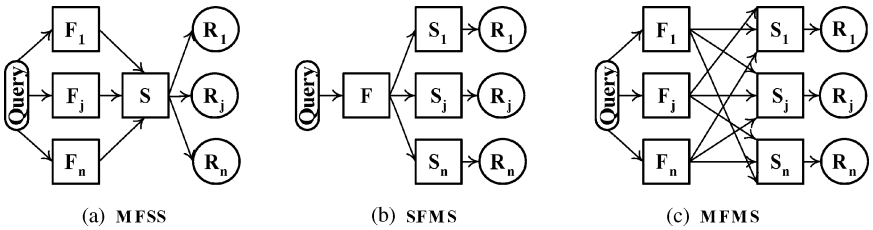


Figure 2. Multiple formulations and multiple schemes.

Information retrieval can be viewed as a process which takes a query (Q) as an input and produces the output which is a list of documents or results (R) (see figure 1(a)). The IR process entails a query formulation (F) or representation and a scheme or system (S) which processes the query formulation in order to obtain results (R) (see figure 1(b)).

With the advent of computer science and information technology (in particular, database technology and information retrieval technology), it has become feasible and possible to improve information retrieval system performance by considering multiple formulations and multiple schemes. Figure 2 depicts three such possibilities.

MFSS—multiple formulations single scheme (figure 2(a)), SFMS—single formulation multiple schemes (figure 2(b)), and MFMS—multiple formulations multiple schemes (figure 2(c)). However, very few of the developments have actually investigated the effect of multiple formulations and/or multiple retrieval schemes on performance.

Saracevic and Kantor (1988) stated explicitly that taking into account the different results of the different formulations could lead to retrieval performance better than that of any of the individual query formulations. The project reported in Belkin et al. (1993) studied the effect of combining multiple representations of information problems on the performance of the INQUERY probabilistic inference network retrieval engine. Although their results showed that, in general, progressive combinations of query formulations lead to progressive improvement in retrieval performance, the INQUERY results (INQC) were substantially better than those of the combined Boolean queries. The authors then considered the issue of combining INQC and the combined Boolean queries as two different sources of evidence. The overall retrieval performance became worse when more weight was given to the Boolean query evidence. However, the performance was improved when fractional weights were given to the combined Boolean queries.

Belkin et al. (1994) and Fox and Shaw (1994) investigated the effect of combination of multiple representations of TREC topics on retrieval performance. Both projects found that the best method of combination often led to results that were better than the best performing single query. However, they indicated that choosing the best query often results in significant performance differences from combined queries. They also pointed out that

in any single run there are always instances of combined queries performing better than the best, and on average combination does better. Belkin et al. (1995) reported on two studies conducted at Rutgers University (Belkin et al. 1994) and at Virginia Tech (Fox and Shaw 1994) that investigated the effect on retrieval performance of combination of multiple representations of TREC-2 topics. When dealing with query combination, the rules used (CombSUM, CombANZ, and CombMNZ) were based on similarity scores between a topic and a document. On the other hand, when dealing with multiple evidences from different schemes (or systems), combinations (MAX, MIN and MED) were based on rank information. Encouraged by the interesting and generally positive results of the two separate studies involving combination of evidences (using similarity scores) or data fusion (using rank information), Belkin et al. (1995) performed two other experiments and had the following observations:

Remark 1.1. (a) When different systems are commensurable, combination using similarity scores is better than combination using only ranks; (b) when multiple systems have incompatible scores, a combination method based on ranked outputs rather than the scores directly is the proper method for combination; and (c) although results from the experiments for combination of results from different databases are encouraging, it is not clear that such combination is possible among systems that have different methods for computing similarity scores.

The paper by Pfeifer et al. (1996) gave a review of known similarity measures in a search for proper names. Their experiments (on measures dealing with phonetic similarity, typing errors, and plain string similarity) showed that all three approaches perform significantly better than a system based on exact-match searches only. They suggested that further improvements are possible by combining different methods. Although they realized that combining two or three different similarity measures seems to be very promising, they indicated that further work for maintaining and searching one or two more methods has to be considered.

Lee (1997) presented the rationale for evidence combination that different runs return similar sets of relevant documents but retrieve different sets of non-relevant documents. He also investigated the effect of using ranks instead of using similarity on retrieval effectiveness. In particular, he showed experimentally that in some circumstances, using ranks works better than using similarity. He also investigated the effect of using rank instead of similarity on retrieval effectiveness and found that:

Remark 1.2. Data fusion using rank works better than using similarity scores if the runs in the combination have 'different' rank-similarity curves.

In their study of the problem of predicting, in advance, whether combination (or fusion) of two or more retrieval schemes will be worth doing, Ng and Kantor (1998) identified:

Remark 1.3. Two predictive variables for the effectiveness of the combination: (a) a list-based measure of output dissimilarity, and (b) a pair-wise measure of the similarity of the performance of the two schemes.

In a subsequent study, Ng and Kantor (2000) investigated the prediction power of these two variables using symmetrical data fusion and receiver operating characteristic (ROC) curve. Using precision at the 100th document, $P_{@100}$, to represent efficacy similarity, they use ratio P_l/P_h (P_l and P_h are $P_{@100}$ for the lower and higher performance schemes respectively) as a variable to measure the similarity of performance of the two IR schemes. Although they found that most of the positive cases have ratio of precision P_l/P_h close to 1, they also stated that the two predictive variables do not completely determine whether simple (linear) and symmetric data fusion will be effective.

The LC (linear combination) model for fusion of IR systems combines the results lists of multiple IR systems by scoring each document with a weighted sum of the scores from each of the component systems. Vogt and Cottrell (1999) studied the problem of predicting the performance of a combined system. Their analysis supports the following:

Remark 1.4. An LC model should only be used when the systems involved have high performance, a large overlap of relevant documents, and a small overlap of non-relevant documents.

Previous empirical and experimental results (including those reviewed in this section) have achieved certain statistical success in understanding the effectiveness of data fusion (with multiple formulations of queries, or multiple schemes, or in different runs) in information retrieval. However, the general questions of "why" and "how" DF in IR can be effective still remain unanswered. All these indicate that the problem involves tremendously high complexity and dimensionality. They have become both quantitatively and qualitatively difficult to trace. In an IR system (see figure 1), different schemes (systems or engines) can use different techniques (or algorithms) to measure the likelihood or probability of relevance of a document to a given query. Moreover, the choices of techniques (or algorithms) rely heavily on the application domain they are applied to or used in. This situation is complicated by having a variety of multiple formulations of the information need and a large and multi-faceted collection of documents (see figure 2(a)–(c)). Multiple representations (or query formulations) can occur either as a result of the interpretation of the original need by multiple experts or as disjoint or non-disjoint subsets from the partition of the original query (such as a long query). In both cases, they also involve semantic consideration. On the other hand, the document space consists of not only large and different structured database systems but also a variety of sites (such as the World Wide Web) located in different networks and different countries.

In this paper, we continue the study of the problem of data fusion (DF) in information retrieval domain (see figure 2). On one hand, we restrict ourselves to information retrieval using similarity measures to search for proper (relevant) documents in the databases or on the World Wide Web when presented with an information need (a query). On the other hand, even though we include the general MFMS setting (see figure 2(c)), we only consider the case of combining results of search in the same database or search space. In general, we have found:

Remark 1.5. (a) Different formulations (or representations) can be derived from the same query by different experts. But they can also be obtained from different (disjoint or

non-disjoint) subsets of the same query; and (b) the search can be based on different formulations (see figure 2(a)) or/and using different schemes (or systems) (see figure 2(b) and (c)) on the same database (or on the World Wide Web).

Data fusion is a process (acquisition, design, and interpretation) of combining information gathered by multiple agents (sources, schemes, sensors or systems) into a single representation (or result). Data fusion has been used in pattern recognition where results from multiple recognizers (or classifiers) with different feature extracts are combined so as to achieve better results (Xu et al. 1992). Multiple sensor DF has been studied in various application domains such as signal detection, target tracking, image processing, surveillance and defense applications (Hsu et al. 2003, Lyons et al. 2003, Varshney 1997).

The concept of data fusion has been used, as mentioned above, in information retrieval to study the combination of multiple evidences resulting from different query formulations or from different schemes (Belkin et al. 1993, 1994, 1995, Fox and Shaw 1994, Kantor 1998, Lee 1997, Ng and Kantor 1998, 2000, Pfeifer et al. 1996). Many empirical studies have been performed and various results have been obtained. While some of the major issues related to the questions such as why and how multiple evidences should be combined remain unanswered, researchers have come to realize the advantage and benefit of combining multiple evidences.

Our approach aims to study the problem of when DF in IR is worth doing and how fusion should be done. We take the modeling approach, which will encompass several fundamental issues related to the theoretic treatment of the complex problem. We establish a model based on Cayley graphs and digraphs (called CG model) with the following characteristics:

Remark 1.6. (a) Each of the multiple evidences (say evidence A) is represented as a ranked list of two functions $(x, r_A(x))$ and $(d, s_A(d))$ indicating ranks with the *rank function* (the document $r_A(x)$ is ranked x), and documents with the *score function* (the document d has similarity score $s_A(d)$) respectively; and (b) assuming that there are n different documents, $r_A(x)$ is then considered as a permutation of these n documents and $s_A(d)$ is a function from the set of n documents to the set of real numbers.

Our model uses a ranked list which consists of a rank function (as a permutation in the set of all permutations of n elements S_n) and a score function (which is the similarity score of the document). We perform analytical study and simulation of the DF of different kinds of ranked lists and investigate the effectiveness of these DF's. We also study DF techniques using rank vs. score combination and explore further the question of when and why one kind of combination is better than the other. We believe that our model and approach will provide better understanding of the phenomena surrounding the issue of effectiveness of DF in information retrieval.

In Section 2, we describe our data fusion framework which includes a data fusion model and architecture of combining two evidences (i.e. two ranked lists). We also give definition of a Cayley graph and introduce the concept of rank/score function. Section 3 gives an analytical result which strongly supports the advantage of using the framework. Experimental results are included in Section 4. More detailed discussions and remarks are summarized in Section 5 which concludes the paper.

2. Data fusion model and architecture

We first review and define some of the notations and terminologies, which will be used in latter sections. For positive integers k and n, let $[n] = \{1, 2, 3, 4, \ldots, n\}$ and $[k, n] = [n] - [k-1]$. Similarly, we define $[d_n]$ to be $\{d_1, d_2, \ldots, d_n\}$. A permutation α on $[n]$ is an one to one mapping from $[n]$ to itself . It can be written as the following different, but equivalent, forms:

x	1	2	3	$\ldots$	n
$\alpha(x)$	$\alpha(1)$	$\alpha(2)$	$\alpha(3)$	$\ldots$	$\alpha(n)$

and

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \ldots n \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \ldots \alpha_n \end{pmatrix} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_n] = [\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4 \ldots \alpha_n]$$

It can also be written as the product of disjoint cycles each consisting of elements from $[n]$:

$$\alpha = (\alpha_{11}\ \alpha_{12} \ldots \alpha_1 k_1)(\alpha_{21}\ \alpha_{22} \ldots \alpha_2 k_2)(\alpha h_1\ \alpha h_2 \ldots \alpha h k_h)$$

where

$$\alpha(\alpha_{ij}) = \alpha_i(j+1),\ \alpha(\alpha_i k_i) = \alpha_{i1} \quad \text{and} \quad \sum_{i=1}^{h} k_i = n.$$

For example, when α is a permutation on the set of numbers $\{1, 2, 3, 4, 5, 6\}$, we have

x	1	2	3	4	5	6
$\alpha(x)$	4	6	3	5	1	2

and

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 3 & 5 & 1 & 2 \end{pmatrix} = [4, 6, 3, 5, 1, 2] = (1\ 4\ 5)(2\ 6)(3) = (1\ 4\ 5)(2\ 6).$$

Often a cycle of length one is ignored without any ambiguity. We also adopt the convention that each permutation is written interchangeably (without confusion) as an ordered list of elements of $[n]$ and as concatenations of cycles of elements of $[n]$. Let S_n be the set of all permutations on $[n]$. Define binary operation "$*$" between two permutations α and β in S_n as $(\alpha * \beta)(x) = \alpha(\beta(x))$. The set S_n together with the binary operator "$*$" forms a group. It is also called the *symmetric group S_n of order n*. We now define the concept of a group and a graph.

Definition 2.1. Let Γ be a finite set of n elements and $*$ be a binary operation in $\Gamma.\Gamma$ is said to be a *group* if it satisfies the following properties:

(a) for every $a, b \in \Gamma$, $a * b \in \Gamma$,
(b) for every $a, b, c \in \Gamma$, $(a * b) * c = a * (b * c)$,

(c) there exists an identity element $\mathbf{e}$ in Γ, such that $\mathbf{e} * a = a * \mathbf{e} = a$ for all a in Γ, and
(d) for every $a \in \Gamma$, there exists b_l and b_r such that $b_l * a = \mathbf{e}$ and $a * b_r = \mathbf{e}$. The two elements b_l and b_r are called the *left inverse* and *right inverse* of the element a respectively.

The two properties in (a) and (b) are called *closure property* and *associativity* respectively. If for any two elements a, b in Γ, $a * b = b * a$, then Γ is said to be *commutative*. Often in this case, Γ is said to be an *Abelian group*.

Definition 2.2. Let V be a set of n elements, E a set of collection of subsets with 2 distinct elements from V, and A a set of collection of ordered pairs with distinct elements from V. For simplicity, we assume the subsets in E (and the ordered pairs in A) are distinct. $G = (V, \mathsf{E})$ and $D = (V, A)$ are said to be a *graph* and a *directed graph* respectively, with E as the edge set of G and A as the arc set of D.

We note that the symmetric group S_n of order n is a special case of a kind of algebraic entity called *permutation group*. For definition and properties of a permutation group, the readers are referred to the book by Biggs and White (1979).

The symmetric group S_n, when imposed a metric, would become a metric space. For example, when the metric is Kendall distance $d_k(\alpha, \beta)$ which counts the number of discordant pairs between α and β, S_n then becomes a metric space denoted as (S_n, d_k). However, since the metric space (S_n, d_k) is discrete and it is a special case of a more general structure, we define the concepts of a Cayley graph and a Cayley digraph as follows:

Definition 2.3. Let Γ (or Γ') be a group and S (or S') a generating set which does not include identity element of Γ (or Γ'). *Cayley digraph* $G(\Gamma, S)$ is the directed graph with node set $V(G) = \Gamma$ and arc set $A(G) = \{(a, b) \mid a, b \in \Gamma, ba^{-1} \in S\}$. The *Cayley graph* $G'(\Gamma', S')$ is an undirected graph with node set $V(G') = \Gamma'$ and edge set $E(G') = \{(c, d) \mid c, d \in \Gamma', cd^{-1}, dc^{-1} \in S\}$.

The study of Cayley graphs and digraphs, sometimes under the name Cayley color-group or Cayley diagrams, can be dated to the 1940's. Recent survey and treatments can be found in Biggs and White (1979) and Grammatikakis et al. (2001), Chap. 6.4, and Heydemann (1997). In our applications here, we are more concerned with the case of Cayley graph where $\Gamma = S_n$, the symmetric group, and S is a generating set with transpositions. Two kinds of S we are most interested in are T_1 and T_2:

$$T_1 = \{(t, t+1) \mid t \in [1, n-1]\},$$

and

$$T_2 = \{(i, j) \mid i, j \in [n], i \neq j, i < j\}.$$

We are using the Cayley graph $G(S_n, T)$, $T = T_1$ or $T = T_2$, as the rank space. $G(S_n, T_1)$ and $G(S_n, T_2)$ are closely related to the two metric spaces (S_n, d_k) (defined by Kendall distance $d_k(\alpha, \beta)$) and (S_n, d_{cay}) (using Cayley's distance $d_{\text{cay}}(\alpha, \beta)$). Both distances d_k and

d_{cay} can be found in Marden (1995). While $d_k(\alpha, \beta)$ counts the number of discordant pairs between α and β, $d_{\text{cay}}(\alpha, \beta)$ counts the minimum number of arbitrary pair-wise interchanges needed to bring the α order $[\alpha_1, \alpha_2, \ldots., \alpha_n]$ to the β order $[\beta_1, \beta_2, \ldots., \beta_n]$.

When $d_k(\alpha, \beta) = 1$, α and β are related to (or incident with) each other by an adjacent transposition τ in S_n. It follows that (S_n, T_1) is a graph where S_n is the set of all permutations of $[n]$, $[\alpha_1, \alpha_2, \ldots.\alpha_n]$, and $T_1 = \{(t, t+1) \mid t \in [1, n-1]\}$ defines the adjacency among two nodes α and β ($\alpha \sim \beta$ if $\beta = \alpha \circ \tau$ for some τ in T_1). In fact, this means that Kendall distance $d_k(\alpha, \beta)$ is equivalent to the graph distance $d(\alpha, \beta)$ calculated in the Cayley graph $G(S_n, T_1)$. The same kind of equivalence occurs when the distance is the Cayley distance and the adjacency in the graph is defined using the sets of transpositions T_2. Since the Cayley distance d_{cay} (α, β) counts the minimum number of arbitrary pair-wise interchanges needed to bring the order of α to the order of β, the pair-wise interchanges are the permutations which are transpositions $T_2 = \{(i, j) \mid i, j \in [n], i < j, i \neq j\}$. Hence the Cayley graph (S_n, T_2) is equivalent to the metric space (S_n, d_{cay}).

We note that although Cayley distance $d_{\text{cay}}(\alpha, \beta)$ and Cayley graph (and Cayley digraph) share the same name "Cayley", they are different in the sense that the former defines a distance between the rankings (or permutation) and the latter defines a graph on a group of permutations.

In our approach of studying DF in information retrieval, a ranked list consists of a rank function and a score function. *A rank function* $r_A(x)$ is a ranking of the documents in $D = \{d_1, d_2, \ldots, d_n\}$, where the document $d = r_A(x)$ is assigned the rank of x. A *score function* $s_A(d)$ is the similarity score assigned to the document d. Although often we use the numerical subindices to denote the documents, it is easy to confuse document ordering and ranking. In order to alleviate this problem, we use $d_1, d_2, \ldots, d_n$ to indicate ordering of the n documents and when there is no confusion, $1, 2, 3, \ldots, n$ are used to mean $d_1, d_2, \ldots, d_n$. Hence the two functions, rank function and score function, ($r_A(x)$ and $s_B(d)$ in Remark 1.6), are listed as follows (when $r_A(x)$ is a function from [10] to $[d_{10}]$ and $s_A(d)$ is a function from $[d_{10}]$ to [0, 1] = the set of real numbers between and including 0 and 1.

x	1	2	3	4	5	6	7	8	9	10
$r_A(x)$	d_2	d_3	d_5	d_6	d_8	d_4	d_{10}	d_7	d_1	d_9

and

d	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
$s_A(d)$	0.1	1.0	0.9	0.4	0.7	0.6	0.2	0.5	0.1	0.3

We are now ready to define the concept of a rank/score function.

Definition 2.4. *The rank/score* function f_A for the system A is a function from $[n]$ to $[0, s] = \{x \in R^+ \text{ and } 0 \leq x \leq s\}$, where s is the highest score the system A can have in the set of non-negative real numbers R^+ such that $f_A(x) = (s_A \circ r_A)(x) = s_A(r_A(x))$ for x in $[n]$.

It follows that f_A has the following values when $n = 10$ in the example above.

x	1	2	3	4	5	6	7	8	9	10
$f_A(x)$	1.0	0.9	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.1

For a ranked list A with $r_A(x)$ and $s_A(d)$ and $q \in [n]$, we define the following two parameters to measure the performance of the system (or scheme) of the ranked list A.

Definition 2.5 (Precision at q and average precision). Let $A_{(k)} = \{r_A(i) \mid i \leq k\}$ and $A^{(k)} =$ smallest set $A_{(j)}$, $1 \leq j \leq n$, s.t. $|A_{(j)} \cap \text{Rel}| = k$, where Rel = set of all documents that are judged to be relevant. If $|\text{Rel}| = q$ for some integer $q (0 < q \leq n)$, we define two measures of performance for a system A as follows:

$$\text{Precision at } q \text{ of } A : P_{@q}(A) = \frac{|\text{Rel} \cap A_{(q)}|}{q}$$

$$\text{Average precision of } A : P_{\text{avg}}(A) = \frac{\sum_{i=1}^{q} \frac{i}{|A^{(i)}|}}{q}$$

For two ranked lists A and B, we present two different ways of combining A and B. One uses rank combination and the other uses score combination.

Definition 2.6 (Rank combination). Given two ranked lists A and B with $r_A(x)$, $s_A(d)$ and $r_B(x)$, $s_B(d)$ respectively, let $g_{AB}(d) = (1/2)[r_A^{-1}(d) + r_B^{-1}(d)]$. Sort the array $g_{AB}(d)$ in ascending order and let $s_g(d)$ be the resulting array. Since the two arrays $s_g(d)$ and $f_g(x)$ are equivalent to each other with $r_C(x) = d$, the ranked list C which is the combination of A and B using ranks has the rank function $r_C(x) = d$ with $f_g(x) = s_g(r_C(x))$.

Definition 2.7 (Score combination). Given two ranked lists A and B with $r_A(x)$, $s_A(d)$ and $r_B(x)$, $s_B(d)$ respectively, let $h_{AB}(d) = (1/2)[s_A(d) + s_B(d)]$. Sort the array $h_{AB}(d)$ in descending order and let $s_h(d)$ be the resulting array. Since the two arrays $s_h(d)$ and $f_h(x)$ are equivalent to each other with $r_D(x) = d$, the ranked list D which is the combination of A and B using scores has the rank function $r_D(x) = d$ with $f_h(x) = s_h(r_D(x))$.

We illustrate the above two definitions with the example in figure 3 for a special case $n = 10$. Note that each of the two rank functions $g_{AB}(x)$ and $h_{AB}(x)$ may contain duplicate values (such as in $g_{AB}(x)$ in figure 3(c)). When this happens, we use the convention of choosing the smaller rank in the inverse mapping g_{AB}^{-1} or h_{AB}^{-1}. Therefore in figure 3(d), we would pick "d_1" first and then "d_8" because $g_{AB}^{-1}(f_g(6)) = g_{AB}^{-1}(6.5) = \{d_1, d_8\}$.

From Definition 2.4, the rank/score function f_A is a function from $[n]$ to $[0, s] = \{x \in R^+$ and $0 \leq x \leq s\}$ and is independent of the ranked or ordered documents. The function $s_A(d)$ is then obtained as $s_A(d) = f_A(r_A^{-1}(d))$. Figures 3(a) and (b) give two examples for $s_A(d)$ and $s_B(d)$.

Figure 4 gives four rank/score functions grouped in two different settings to show the contrast. The two functions in figure 4(a) are taken from the two score functions in figures 3(a) and (b). The two examples in figure 4(b) have $n = 500$ and $s = 100$.

x	1	2	3	4	5	6	7	8	9	10
$r_A(x)$	d_2	d_8	d_5	d_6	d_3	d_1	d_4	d_7	d_{10}	d_9
$s_A(d)$	10	7	6.4	6.2	4.2	4	3	2	1	0

(a) Ranked list A

x	1	2	3	4	5	6	7	8	9	10
$r_B(x)$	d_5	d_9	d_6	d_2	d_8	d_7	d_1	d_3	d_{10}	d_4
$s_B(d)$	10	9	8	7	6	5	4	3	2	1

(b) Ranked list B

d	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
$g_{AB}(d)$	6.5	2.5	6.5	8.5	2	3.5	7	3.5	6	9

(c) Combinations of A and B by rank

x	1	2	3	4	5	6	7	8	9	10
d	d_5	d_2	d_6	d_8	d_9	d_1	d_3	d_7	d_4	d_{10}
$s_g(d)$:$f_g(x)$	2	2.5	3.5	3.5	6	6.5	6.5	7	8.5	9
$r_C(x)$	d_5	d_2	d_6	d_8	d_9	d_1	d_3	d_7	d_4	d_{10}

(d) Sorted scores $g_{AB}(d)$ into $s_g(d)$ with document indices

d	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
$h_{AB}(d)$	4.0	8.5	3.6	2.0	8.2	7.1	3.5	5	4.5	1.5

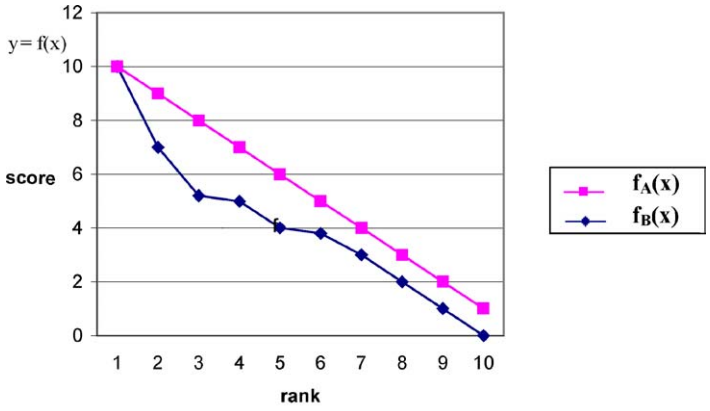
(e) Combinations of A and B by score

x	1	2	3	4	5	6	7	8	9	10
d	d_2	d_5	d_6	d_8	d_9	d_1	d_3	d_7	d_4	d_{10}
$s_h(d)$:$f_h(x)$	8.5	8.2	7.1	5.0	4.5	4.0	3.6	3.5	2.0	1.5
$r_D(x)$	d_2	d_5	d_6	d_8	d_9	d_1	d_3	d_7	d_4	d_{10}

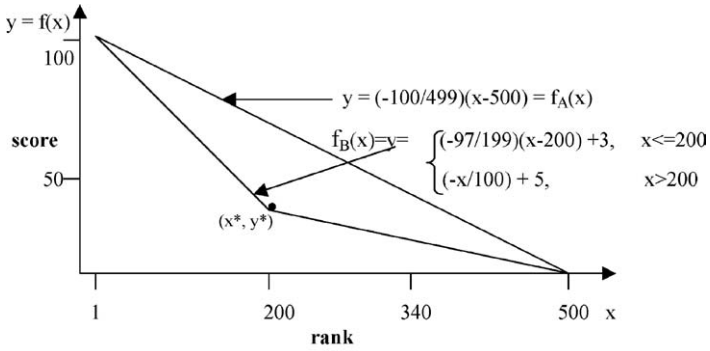
(f) Sorted scores $h_{AB}(d)$ into $s_h(d)$ with document indices

Figure 3. Combinations using rank vs. score, $n = 10$.

Our approach to the study of effectiveness of DF in IR consists of a model of simulation and analysis and an architecture summarized in figure 5. We use the symmetric group S_n as our sample space (or sometimes called rank space) with respect to n documents. Since the total number of possible rank data written as permutations is n! which is computationally intractable, we use the diagram in figure 5 to simulate the phenomena. Two basic rankers (or ranked lists) are used (called A and B). Ranker A has a rank function $r_A(x)$, a score function $s_A(d)$, and the performance P(A), $P_{@q}$(A) or P_{avg}(A). Ranker B is represented in the same fashion. Ranked lists C and D are rank combinations and score combinations of A and B respectively as defined in Definitions 2.6 and 2.7. By employing different variations of A



(a) Two rank/score functions with n=10, s=10



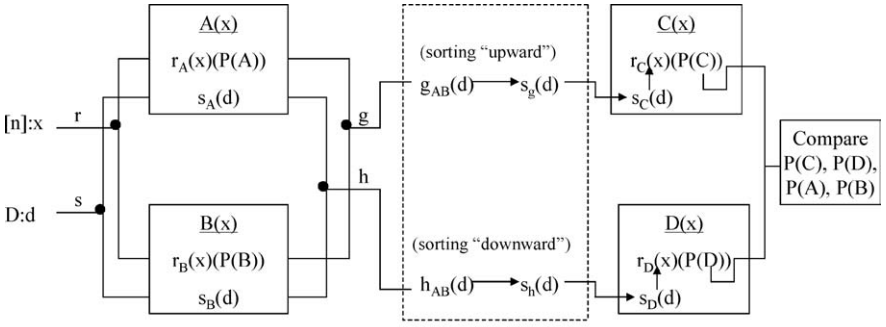
(b) Two rank/score functions with n=500 and s=100

Figure 4. Four rank/score functions in two different groups.

and B, we hope to be able to extend and generalize our results. In Section 4, we will have results of our simulation in two cases. The first case deals with the situation where $r_A(x)$ is fixed as the identity permutation and $f_A(x)$ is also fixed as a straight line passing through points $(1, s)$ and $(n, 0)$ in the rank/score function graph. In the second case, $f_A(x)$ is fixed as in the first case, but $r_A(x)$ is obtained as a random permutation. In the next section (Section 3), we will show that when $r_B = t \circ e_A$, the composition of $t \in T_2$ and the identity function e_A, $f_B(x)$ has one single turning point (a, b), and $q < a$ with certain conditions, then the performance of the combination by ranks is always better than that of the combination by score, i.e. $P_{@q}(C) \geq P_{@q}(D)$.

3. Analysis of combination methods

In this section, we take the general view as stated in Remark 1.6. As such, each evidence A is presented as two functions $r_A(x)$ and $s_A(d)$ indicating ranks with ranked documents and documents with their similarity scores for n distinct documents. Each rank function



$$\begin{cases} g_{AB}(d) = \dfrac{1}{2}[r_A^{-1}(d) + r_B^{-1}(d)] \\ h_{AB}(d) = \dfrac{1}{2}[s_A(d) + s_B(d)] \end{cases}$$

$$\begin{cases} r_C(x) = d \text{ and } f_g(x) = s_g(d), \text{(hence } f_g(x) = s_g(r_C(x)) \text{ or } r_C^{-1}(d) = f_g^{-1}(s_g(d))) \\ r_D(x) = d \text{ and } f_h(x) = s_h(d), \text{(hence } f_h(x) = s_h(r_D(x)) \text{ or } r_D^{-1}(d) = f_h^{-1}(s_h(d))) \end{cases}$$

Figure 5. DF architecture.

$r_A(x) = [A_1, A_2, A_3, \ldots, A_n]$ is then considered as a permutation of the n documents (or objects in general). Therefore, $r_A(x)$ is considered an element in the rank space S_n and a node in the Cayley digraph $G(S_n, T_1)$ (see Definition 2.3 and definition of T_1). Armed with the framework described in Section 2 and previous results discussed in Section 1, we are now able to formulate the central problems in the study of data fusion in information retrieval domain. Let A and B be two evidences presented as $[n]$, $r_A(x)$, $s_A(d)$ and $[n]$, $r_B(x)$, $s_B(d)$ respectively. These are also considered as nodes in the Cayley digraph $G(S_n, T)$ for some T. Let C and D be the results of fusion from A and B defined in Definition 2.6 and Definition 2.7 respectively. Let $P(C)$ and $P(D)$ be the performance measurement defined in Definition 2.5 (see also figure 5). We summarize Remarks 1.1–1.6 and ask the following questions:

Remark 3.1. For what A and B, $P(C)$ (or $P(D)$) $\geq \max\{P(A), P(B)\}$ and for what A and B, $P(C) \geq P(D)$?

Since the rank space S_n has $n!$ elements, the number of possible (A, B) pairs is of order $(n!)(n! - 1)/2 = O((n!)^2)$ which is computationally unmanageable. In the following section (Section 4), we will study the problem for two different cases. In particular, we will investigate in Section 4 by simulation the performance of C (and of D) for two cases: *Case 4.1*: $r_A = e_A$ the identity permutation and r_B = random, and *Case4.2*: r_A = random and r_B = random. Actual values we used are $n = 500$, $s = 100$ with precision at 50 ($P_{@50}$) and average precision (P_{avg}). Among the many results and phenomena observed from these simulations, we see the following pattern:

Remark 3.2. Let A and B be represented as r_A, s_A and r_B, s_B respectively. Let C and D be obtained and represented as r_C, s_C and r_D, s_D respectively as in figure 5 in Section 2. As long as f_A and f_B are "far apart" and $q \sim n/10$, then $\%(P_{@50}(C) > P_{@50}(D)) >> \%(P_{@50}(C) < P_{@50}(D))$ and $\%(P_{\text{avg}}(C) > P_{\text{avg}}(D)) >> \%(P_{\text{avg}}(C) < P_{\text{avg@50}}(D))$ in the case when $r_A = e_A$ and $r_B =$ random. In the cases that $r_A =$ random and $r_B =$ random, $\%(P_{\text{avg}}(C) > P_{\text{avg}}(D)) > \%(P_{\text{avg}}(C) < P_{\text{avg}}(D))$.

We now analyze the special case when $r_A = e_A$, the identity permutation, and $r_B = t \circ e_A$, where $t \in T_2$, and f_A, f_B are two non-increasing functions. We assume that f_A is the straight line $L((1, s), (n, 0))$ connecting the two end points $(1, s)$, $(n, 0)$ and f_B is the combination of the two straight lines with end points $L_1((1, s), (x, y))$ and $L_2((x, y), (n, 0))$ which meet at (x, y). We state and prove the following theorems (see figure 4(b) in Section 2 for the special cases $n = 500$ and $s = 100$):

Theorem 1. *Let A, B, C and D be defined as before. Let $f_A = L$ and $f_B = L_1 \, U \, L_2$(L_1 and L_2 meet at (x^*, y^*)) be defined as above. Let $r_A = e_A$ be the identity permutation and $r_B = t \circ e_A$, where $t \in T_2$ and $t = (i, j)$. If $q < x^*$ and (a) $i < j < q$, (b) $q < i < j$, (c) $i < q < j < x^*$, or (d) $i < q < x^* < j$, where max $\{h_{AB}(i), h_{AB}(j)\} \leq y^+ = (1/2)[y^* + f_A(x^*)]$ and $(1/2)(i + j) > x^*$, then $P_{@q}(C) \geq P_{@q}(D)$.*

See Appendix A for proof.

Theorem 2. *Let $A, B, C, D, r_A, f_A, r_B$ and f_B be defined as in Theorem* 1. *If $q < x^*$ and either (a), (b), (c) or (d) in Theorem* 1 *is satisfied, then $P_{\text{avg}}(C) \geq P_{\text{avg}}(D)$.*

Proof is similar to that of Theorem 1.

4. Simulation

In this section, we describe the simulation results for different cases. In each of the cases, we assume the number of documents to be $n = 500$ and the highest score given to any rank is $s = 100$. Hence the total number of possible permutations as rank function is 500!. The rank functions rB's are obtained by a random generation process in Case 4.1. In each simulation, we generate ten thousand (10 k) cases of r_B. In our study, we fix f_A to be the straight line connecting the two end points (500, 0) and (1, 100). Since f_B can be any discrete function defined from [1, 500] to [0, 100] which is monotonically non-increasing, we start with a special case of f_B which is a combination of two straight lines with one turning point (x^*, y^*). Note that the point (200, 30) is such a turning point for the rank/score function f_B in figure 4(b) in Section 2. On the other hand, the rank/score function f_B in figure 4(a) has no such points. Since the problem at issue is combining two ranked lists A and B, we would like to see r_A and r_B as arbitrary as possible. Therefore we include a case where r_A and r_B are both randomly generated in Case 4.2. These two cases are described in more details as follows:

Case 4.1 ($r_A = e_A$ the identity permutation, $r_B =$ random). In this case, $r_A = e_A$, f_A is the straight line connecting (500, 0) and (1, 100). In fact, f_A has the following formula y

$= (-100/499)\ (x - 500)$ (See f_A in figure 4(b) in Section 2). For each permissible turning point (x^*, y^*) for the rank/score function f_B we generate 10 k r_B's. Then we combine these 10 k ranked lists B's with ranked list A. The results are listed in figure 6 using $P_{@50}$ and P_{avg} respectively. In figure 6(a), the tuples at point (x^*, y^*) (i.e. (a, b, c)) where a, b and c are the number of cases out of the 10 k cases so that $P_{@50}(C) < P_{@50}(D)$, $P_{@50}(C) > P_{@50}(D)$ and $P_{@50}(C) = P_{@50}(D)$ respectively. Likewise, figure 6(b) exhibits the values of (a, b, c) in percentages (out of the 10 k cases) with one decimal point. Figure 6(c) uses P_{avg} instead of $P_{@50}$. In these cases, only values of (a, b) are used as it rarely happens that $P_{\text{avg}}(C) = P_{\text{avg}}\ (D)$.

Case 4.2 ($r_A =$ random, $r_B =$ random). In this case, f_A is the same straight line as in Case 4.1 and f_B has the turning point (x^*, y^*). Everything else is the same. We list the results in figure 7(a)–(c).

We note that figures 6 and 7 exhibit certain features which are quite noticeable. One of the most interesting phenomena is that when the turning point (x^*, y^*) is below the standard line (i.e. f_A) at certain locations, the performance of the combination using rank ($P(C)$) is most likely to be better than that of the combination using score ($P(D)$). In the case when r_A is the identity permutation, the results are fairly consistent. Even when r_A is randomly generated (figure 7), $P_{\text{avg}}\ (C)$ is greater than $P_{\text{avg}}\ (D)$ in most of the locations (for turning point for f_B). This prompts us to explore the problem, once again, of finding any other predictive variable for the effectiveness of data fusion. In fact, in the previous section (Section 3), we have shown that under the condition that f_B has the turning point (x^*, y^*) and r_B is the single cycle of permutation (i.e. the transposition (i, j) for any $1 \le i, j \le n$ and $i \ne j$), the combination by rank performs better than combination by score in either $P_{@q}$ or P_{avg} cases as long as $q < x^*$.

We are also interested in the performance of C and D as compared to those of A and B. The data fusion model and architecture we established in Section 2 in this paper is very helpful in the study of data fusion in information retrieval. The simulation procedure is fairly easy to implement. For example, in the quest to find predictive variables for the effectiveness of data fusion, the two predictive measures identified by Ng and Kantor (1998, 2000) are P_l/P_h and $d_k\ (A, B)$ where $P_l = \min\{P(A), P(B)\}$ and $P_h = \max\{P(A), P(B)\}$. In this paper, we have shown analytically and experimentally that the graphical relation between the rank/score functions f_A and f_B, $d\ (f_A, f_B)$, is an indicator to distinguish P(C) and P(D). The data generated and exhibited in figures 8 and 9 demonstrated that the two parameters $d(f_A, f_B)$ and P_l/P_h are, to great extent, barometers to predict the effectiveness of combinations.

Figure 8 lists the change of (a, b, c) along the change of the turning point (x^*, y^*), where $P = P_{@50}$, r_A and r_B are randomly generated, and

$$\begin{cases} a = \text{number of 10 k cases with } P(C) > \max\{P(A), P(B)\}, \\ b = \text{number of 10 k cases with } P(D) > \max\{P(A), P(B)\}, \text{ and} \\ c = \text{number of 10 k cases with } \min\{P(C), P(D)\} > \max\{P(A), P(B)\}. \end{cases}$$

Figure 9 shows the distribution in percentage of the 10 k cases at (x, y), where $x = 0.1$ to 1.0 in step of 0.1 and $y = (50, 10)$ to $(450, 90)$ in steps of $(50, 10)$ and

$$\begin{cases} a = \% \text{ of 10 k cases with } P(C) > \max\{P(A), P(B)\}, \\ b = \% \text{ of 10 k cases with } P(D) > \max\{P(A), P(B)\}, \text{ and} \\ c = \% \text{ of 10 k cases with } \min\{P(C), P(D)\} > \max\{P(A), P(B)\}. \end{cases}$$

Figures 9(a)–(d) deal with $P_{@50}$ and P_{avg} respectively. All these figures have P_l/P_h as the x-coordinate.

5. Discussion and future work

In this paper, we have established a framework (see figure 5) for analysis and simulation in the study of data fusion in the information retrieval domain by defining rank function and score function and using the concept of a rank/score function. Every evidence (from query formulation, retrieval schema or system) is represented as a ranked list (such as A) with three functions: $r_A(x) =$ *rank function*, $s_A(d) =$ *score function* and $f_A =$ *rank/score function*. The rank function r_A is viewed as a permutation of $[d_n]$ = the set of n documents. Using the concept of a Cayley graph, we consider a rank function r_A (of n documents) as a node (and a permutation of $[n]$) of the Cayley graph, (S_n, T), where S_n is the symmetric group of order n and T is a generating set of S_n excluding the identity permutation e.

Recall from Remark 1.6, Definition 2.4 and figure 5, rank function r_A and score function s_A are defined respectively from N to D and from D to R^+. Hence the rank/score function is obtained as $f_A = s_A \circ r_A$. In some application domains, the rank function r_A^* may be defined as the inverse function of r_A (i.e.: from D to N). In such a case, the rank/score function would be $f_A^* = s_A \circ r^{*-1}$ (i.e.: $f_A^* \circ r_A^* = s_A$).

Our current study is the first of a series of investigations exploring the central question of why and how data fusion (or evidence combination) should be done. We have started with some specific cases when the rank/score function $f_A =$ straight line and $f_B =$ semi-linear with one point of intersection (x^*, y^*) (see Sections 3 and 4) even though both functions can be any discrete function defined from [1, 500] to [0, 100] which is monotonically non-increasing (see figure 4(a)). In Section 5.3(d), we will discuss that the condition $n = 500$ can be relaxed to include any constant n. We have proved in Section 3 that if $r_A = \mathrm{e}_A$, the identity permutation, $r_B = t \circ e_A$ where $t \in T_i$ and $q < x^*$ with certain conditions, then $P_{@q}(C) \geq P_{@q}(D)$. Then in Section 4, applying both cases (i) $r_A = e_A$, $r_B =$ random, and (ii) r_A, r_B are random to all 81 points of intersection (x^*, y^*) and generating ten thousand (10 k) permutations for each random case, we have found several interesting phenomena. All these analytical and simulation results, summarized in Sections 5.1 and 5.2, strongly support those findings observed by previous researches surveyed in Section 1 as highlighted in Remarks 1.1–1.4. Section 5.3 discusses our future work on several directions as suggested in the current study.

5.1. *Combination using rank vs. score*

The thrust of our approach is that we are able to define and extract the rank/score function f_A from a ranking procedure A which gives the rank function r_A and score function s_A. On the other hand, the score function s_A can be obtained as $s_A(d) = (f_A \circ r_A^{-1})(d) = f_A(r_A^{-1}(d)) = f_A(x)$ if r_A and f_A are known, where $r_A(x) = d$, d is a document ranked by r_A as rank order x. This differentiation between $f_A(x)$ (defined on ranks) and $s_A(d)$ (defined on documents) enables us to characterize different ranking procedures (algorithms or systems), and then to better quantify the differences between them (see Remarks 1.1 and 1.2). Our results in Sections 3 and 4 with respect to (x^*, y^*), $(y^* < -\frac{1}{5}x^* + 100)$, $(n = 500, s = 100, q = 50)$ confirmed the observations made by previous researchers and summarized in Remark 1.1 (see Belkin 1994, 1995) and Remark 1.2 (see Lee 1997). Specifically, we have demonstrated in our simulation that when $\sum_{x=1}^{500} |(f_A(x) - f_B(x)|$ is big enough, combination using ranks performs better than combination using scores under certain conditions. In particular, we have shown analytically in Theorems 1 and 2 that when the difference between r_A and r_B is a transposition (i, j) with certain conditions and $q < x^*$ with certain conditions, the performance of rank combination is at least as good as that of score combination.

5.2. *Effectiveness of combination*

Various techniques and experiments have been performed to study the effectiveness of combining two or more systems (formulations, algorithms, or different runs) (Aslam et al. 2003, Belkin 1993, 1994, 1995, Hsu et al. 2003, Lee 1997, Lyons et al. 2003, Marden, 1995, Ng and Kantor 1998, 2000, Vogt and Cottrell 1999). These include the progressive combination of query formulations and the linear combination (LC) model for fusion of IR system by scoring each document with a weighted sum of the scores from each of the component systems (Vogt and Cottrel 1999) and the study by Ng and Kantor (1998, 2000) which identified two predictive variables: the Kendall distance and the performance ratio (see Remarks 1.3 and 1.4). The Kendall distance $d_K(r_A, r_B)$ measures the degree of concordance between two different rank lists r_A and r_B. The performance ratio P_l/P_h measures the similarity of performance of the two IR schemes A and B. Our simulation results (see figures 9(a)–(c)) are in conformity with those by Ng and Kantor on the performance ratio P_l/P_h. We have run ten thousand random cases for each of the nine points of intersection (x^*, y^*), where $(x^*, y^*) = (50\text{ t}, 10\text{ t})$ and $1 \le t \le 9$ (see figures 9(a)–(d)).

When considering the positive fusion cases of the combination of different rank lists A and B, the distribution of the positive cases is clustered around $P_l/P_h \sim 1$ for each of the three comparisons regarding effectiveness of the combinations: $P(C)$ vs. $\max\{P(A), P(B)\}$, $P(D)$ vs. $\max\{P(A), P(B)\}$, and $\min\{P(C), P(D)\}$ vs. $\max\{P(A), P(B)\}$, where C and D are combination of A and B using rank and score respectively. As to the Kendall distance $d_K(r_A, r_B)$, we have not attempted to find such pattern in our simulation. However, the simulation results for Case 4.2 discussed in Section 4 and exhibited in figure 8 demonstrated that the graphical behaviors of the rank/score function might be a feasible predictive variable for the effectiveness of combination.

5.3. *Future work*

We have discussed, in Section 4.1, that when $r_A = e_A$ and r_B = random we have $P(C) > P(D)$ (either @50 or on average) (see figures 6(a)–(c)) for most of the cases at point of intersection (x^*, y^*), where $(y^* < -\frac{1}{5}x^* + 100)$. It is interesting to note that when $(y^* > -\frac{1}{5}x^* + 100)$, the situation changes and in fact it becomes the opposite. At the points (x^*, y^*) of intersects where $(y^* = -\frac{1}{5}x^* + 100)$, the situation varies and in majority of the 10 k cases $P(C) = P(D)$ when performance@50 is used. When r_A and r_B are generated at random, slightly higher percentage of the 10 k cases have $P_{\text{avg}}(C) > P_{\text{avg}}(D)$ than $P_{\text{avg}}(D) > P_{\text{avg}}(C)$ (see figure 7(c)). However, when performance@50 is used, no apparent pattern can be drawn (see figure 7(a) and (b)).

The current study suggests several problems worthy of further study and several issues that require further investigations. We summarize as follows:

(a) Let $G_{@q}(X) = P_{@q}(X) - \max\{P_{@q}(A), P_{@q}(B)\}$, where $X = C$ or D. Let $G_{@q}(C,D) = P_{@q}(C) - P_{@q}(D)$. $G_{\text{avg}}(X)$ and $\text{G}_{\text{avg}}(C, D)$ are defined in a similar fashion. In this paper, we have studied the behavior of these parameters under the condition that f_A is linear and f_B is semi-linear with one point of intersection. One direction to pursue is to study the two parameters $\text{G}_{@q}(X)$ and $G_{@q}(C, D)$ when f_A is linear and f_B is piecewise-linear with k points of intersection, or the more general cases, when f_A and f_B are piecewise linear or are in more general situation of being non-increasing monotonic functions.

(b) In our computation of s_C and s_D, we simply take the average of the ranks and scores of A and B respectively (see Definitions 2.6 and 2.7). However, different weights can be assigned to each individual schema and different ways of combinations can be performed in the combination of two or more schemas. Several authors (see Dwork et al. 2001, Fagin et al. 2003, Hsu and Palumbo 2004, Ibraev et al. 2001, Kantor 1998 and Vogt and Cottrell 1994) studied the effectiveness of different weighting assignments and different methods of combination. Our goal in this direction is to extend our results to the weighted combination for A and B (assigning weights α and $1 - \alpha$ to A and B respectively, where $0 < \alpha \leq 1$) and for more than two schemas. In (2004), Hsu and Palumbo studied data fusion in the Cayley graph $G(S_n, T_1)$ to combine A and B using weights α at an increment of 0.1. We also aim to extend our results to compare rank vs. score combinations using different methods of combination such as Markov chain or other non-linear methods.

(c) The current paper has defined the rank/score function f_A (for a schema A) and established an abstract sample space S_n (for a schema A with rank list $r_A = [A_1, A_2, \ldots, A_n]$ on the set of n documents) (our examples use $n = 500$). We have observed (figure(9)(a)–(d)) that positive cases exist when P_l/P_h is close to 1, but have not yet attempted to find any correlation between positive cases and the metric $d_K(r_A, r_B)$ (the Kendall distance). Kantor (1998) has proposed a geometric model which treats P_l, P_h and P_{ideal} (a perfect solution) as three points in an abstract space. Then Ibraev et al. (2001) showed that in the ideal case, the performance of data fusion for a pair IR schemas may be approximated by a quadratic polynomial. From the equation of the curve, it follows that for effective DF the weight of the better schema must be greater than that of the worse schema.

However, some anecdotal evidence suggest that there exist cases where DF is effective when the worse schema has more weight. In our study of DF effectiveness, we can use the rank space S_n with $d_K(r_A, r_B)$ as the distance function. In fact, we can restrict our space to the hyperplane of S_n consisting of all points r_A's $= [A_1, A_2, \ldots, A_n]$ with $\sum_{i=1}^{n} A_i = \frac{n(n+1)}{2}$. We will investigate DF effectiveness using our Cayley graph model (S_n,T_1) and $d_K(r_A, r_B)$ in the hyperspace of $\sum_{i=1}^{n} A_i = \frac{n(n+1)}{2}$ and compare our results with the geometric model studied by Kantor et al (2001) and Kantor (1998). Work along this line has been performed by Hsu and Palumbo (2004) with respect to using S_n as the rank space and Γ (S_n, T_1) as the Cayley graph model. While Kantor et al's approach is considered as a geometric model, the approach of Hsu and Palumbo (2004), and Hsu et al. (2002) using the Cayley graph (S_n, T_1) as a rank space is rather combinatorial.

(d) We note that in our simulation we use $n = 500$. However this condition could be relaxed to include any constant n. The cut of value for precision was chosen to be $q = 50$ which is 10% of the $n = 500$. This ratio is very much related to the real situation of information retrieval systems. We also note that in the real situation the range for s_A and s_B may vary. It is important that these two functions have to be normalized to some common range (in our case, we use $s = 100$ or 1.00) before they can be combined to generate C or D. In general, normalization of the score functions of two or more schemas is a vital step and should have a great impact on the effectiveness of the combinations.

(e) We note that framework proposed and results obtained in this paper for information retrieval can be applied in other domains also. The rank and fuse (RAF) approach for target tracking in CCTV surveillance (Hsu et al. 2003, Lyons et al. 2003) and rank and combine (RAC) method for microarray and gene expression data analysis in bioinformatics (Chuang et al. 2004, Hsu and Palumbo 2004) are two examples of this application.

Appendix A

Proof of Theorem 1

We divide the problem into three cases according to the relative positions of i and j with respect to q: (a) $i < j < q$, (b) $q < i < j$, and (c) $i < q < j$. In the first two cases, it is easy to see that $P_{@q}(C) = P_{@q}(D)$, since the swap of i and j (when i, j are less than or greater than q) does not make any difference to the performance of C or D. Therefore, we consider only case (c) from now on where $i < q < j$. Since $q < x^*$, we then divide case (c) into two subcases:

Subcase (c) (i): $i < q < j < x^*$. We have $s_A(j) - s_B(j) > s_A(i) - s_B(i)$ since the decrease of $s_B(x)$ is faster than that of $s_A(x)$.

Subcase (c) (ii): $i < q < x^* < j$. In this case, $s_A(j) - s_B(j)$ can be greater than, equal to, or less than $s_A(i) - s_B(i)$ depending on how big j is.

In order to prove these two cases, we treat r_A, r_B, s_A, s_B, r_C, r_D, and other related functions or permutations as arrays on the index set $[n]$. Then we have $r_A(i) = i, r_A(j) = j$ and $r_B(i) = j, r_B(j) = i$. Hence we have $g_{AB}(i) = 1/2(i + j) = g_{AB}(j)$. After sorting the

array f_{AB} into ascending order to become s_g, we have

$$s_g\left(\left\lfloor \frac{i+j}{2} \right\rfloor\right) = \frac{1}{2}(i+j) = s_g\left(\left\lceil \frac{i+j}{2} \right\rceil\right).$$

Therefore $r_c(\lfloor \frac{i+j}{2} \rfloor) = i$ and $r_c(\lceil \frac{i+j}{2} \rceil) = j$. The values for $h_{AB}(i)$ and $h_{AB}(j)$ can be calculated using formula (3) and (4).Then the question is: which of the two numbers $h_{AB}(i)$ and $h_{AB}(j)$ is bigger than the other?

In Subcase (c) (i), with $s_A(j) - s_B(j) > s_A(i) - s_B(i)$, we have $h_{AB}(j) > h_{AB}(i)$ by Definition 2.7. After sorting the array h_{AB} into descending order to become s_g, we have $s_h(i') = h_{AB}(j)$ and $s_h(j') = h_{AB}(i)$ for some i', j' in $[i, j]$ with $i < i'$ and $j' < j$. Hence we have $r_D(i') = j$ and $r_D(j') = i$. Hence we have the following situation for subcase (c) (i):

$$\begin{array}{ll}
[n]: & 1, 2, 3, \ldots\ldots, i, \ldots\ldots\ldots, i', \ldots, \lfloor \frac{i+j}{2} \rfloor, \lceil \frac{i+j}{2} \rceil, \ldots, j', \ldots, j, \ldots, x^*, \ldots, n \\
r_C(x): & d_1, d_2, d_3, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, d_i, \quad d_j, \ldots\ldots\ldots\ldots\ldots\ldots\ldots, d_n \\
r_D(x): & d_1, d_2, d_3, \ldots, r_D(i), \ldots, d_j, \ldots\ldots\ldots\ldots\ldots, d_i, \ldots r_D(j), \ldots\ldots\ldots, d_n
\end{array}$$

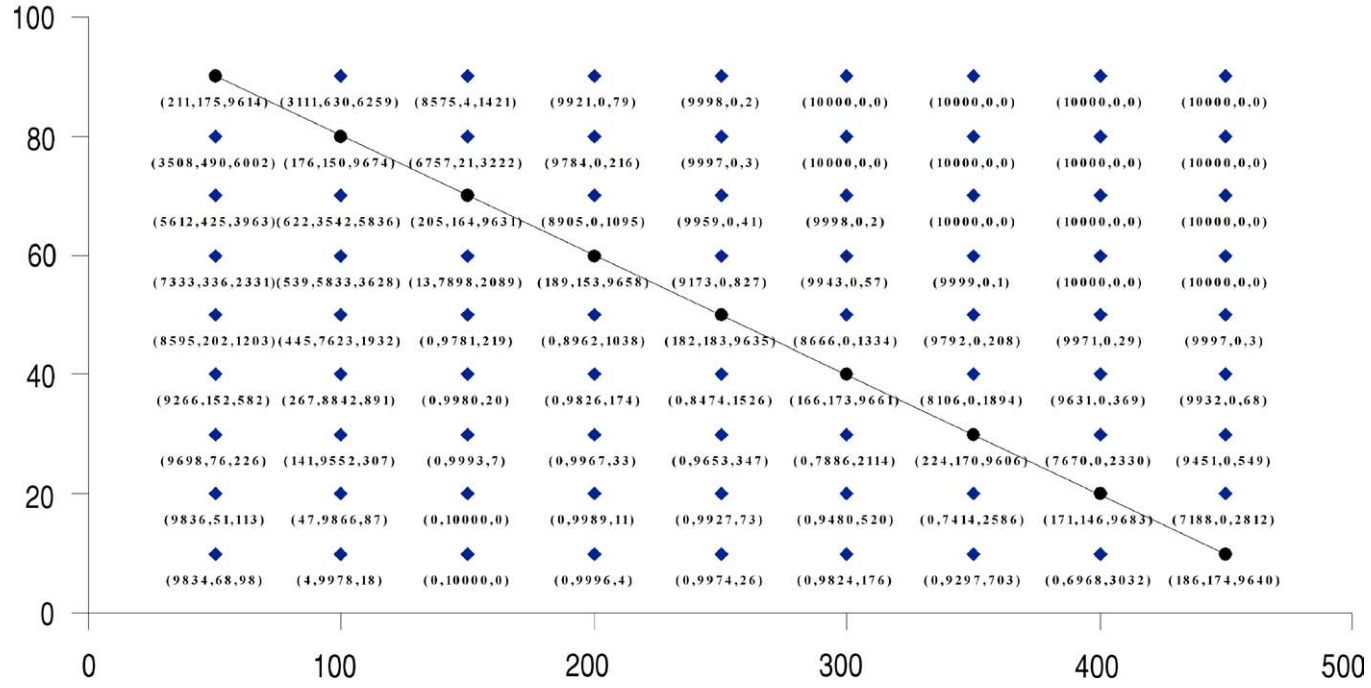
Recall that we have $i < q < j$ in this case. If $q \in ([i, i']U[j', j])$, the theorem holds because $P_{@q}(C) = P_{@q}(D)$. If $q \in ([i', \lfloor \frac{i+j}{2} \rfloor] \cup [\lceil \frac{i+j}{2} \rceil, j])$, then $P_{@q}(C) > P_{@q}(D)$.

In Subcase (c) (ii) where $i < q < x^* < j$, we have three possibilities depending on $s_A(j) - s_B(j)$ is greater than, equal to, or less than $s_A(i) - s_B(i)$. When $s_A(j) - s_B(j) > s_A(i) - s_B(i)$, we have $h_{AB}(j) > h_{AB}(i)$. Hence the proof is similar to previous case, Subcase (c) (i). When $s_A(j) - s_B(j) = s_A(i) - s_B(i)$, we have $h_{AB}(j) = h_{AB}(i)$ by Definition 2.7. It follows that $r_D(i') = d_i$ and $r_D(i' + 1) = d_j$ where $i' \in (i, j)$ and $i < i' < j$. In this case, no matter where q is, we have $P_{@q}(C) \geq P_{@q}(D)$. For the last possibility where $s_A(j) - s(B) < s_A(i) - s_B(i)$, we have $h_{AB}(j) < h_{AB}(i)$ by Definition 2.7. It follows that $r_D(i') = d_i$ and $r_D(j') = d_j$ where $i < i' < j' < j$. Hence we have the following situation for the third possibility of Subcase (c) (ii):

$$\begin{array}{ll}
[n]: & 1, 2, 3, \ldots i, \ldots . i', \ldots, \lfloor \frac{i+j}{2} \rfloor \lceil \frac{i+j}{2} \rceil, \ldots j', \ldots j, \ldots\ldots\ldots n \\
r_C(x): & \ldots\ldots\ldots\ldots\ldots\ldots, d_i, \ldots, d_j, \ldots\ldots\ldots\ldots\ldots\ldots \\
h_{AB}(x): & \ldots\ldots, h_{AB}(i), \ldots\ldots\ldots\ldots\ldots\ldots\ldots, h_{AB}(j), \ldots\ldots \\
s_h(d): & \ldots\ldots\ldots\ldots, h_{AB}(i), \ldots\ldots\ldots\ldots, h_{AB}(j), \ldots\ldots\ldots\ldots\ldots \\
r_D(x): & \ldots\ldots, r_D(i), \ldots, d_i, \ldots\ldots\ldots\ldots\ldots, d_j, \ldots, r_D(j), \ldots\ldots
\end{array}$$

Since in this subcase $h_{AB}(i) > h_{AB}(j)$, $i < i' < j' < j$ and $i < q < x^* < j$, we have either $i' < x^*$ or $i' > x^*$. Let (x^*, y^+) be the middle point between (x^*, y^*) and $(x^*, f_A(x^*))$. By the assumption (d) in the Theorem 1 that $\max\{h_{AB}(i), h_{AB}(j)\} = h_{AB}(i) \leq y^+$, we have $r_D(i') = d_i$, where i' is such that $s_h(i') = h_{AB}(i)$ and $i' > x^*$. On the other hand we have $\frac{i+j}{2} > x^*$. Combining these two inequalities $i' > x^*$ and $\frac{i+j}{2} > x*$ (note: $r_C(\frac{i+j}{2}) = d_i$) together with the assumption $q < x^*$ in this case, we have $P_{@q}(C) = P_{@q}(D)$. This completes the proof of the theorem.

Appendix B



(a)

Figure 6. (a): $r_A = e_A, r_B =$ random. $f_A =$ line between (0, 100) and (500, 0), $f_B =$ single turning point at (x^*, y^*). $P_{@50}$ at (x^*, y^*), $P_{@50}(C)$ vs $P_{@50}(D)$ (<, >, =)—number of cases (total 10, 000), (b): $r_A = e_A$, $r_B =$ random. $f_A =$ line between (0, 100) and (500, 0), $f_B =$ single turning point at (x^*, y^*). $P_{@50}$ at (x^*, y^*), $P_{@50}(C)$ vs $P_{@50}(D)$ (<, >, =)—percentages (number of cases—10, 000), (c): $r_A = e_A$, $r_B =$ random. $f_A =$ line between (0, 100) and (500, 0), $f_B =$ single turning point at (x^*, y^*). P_{avg} at (x^*, y^*), $P_{\text{avg}}(C)$ vs P_{avg}(D). (<, >) − percentages (number of cases—10, 000).

(*Continued on next page.*)

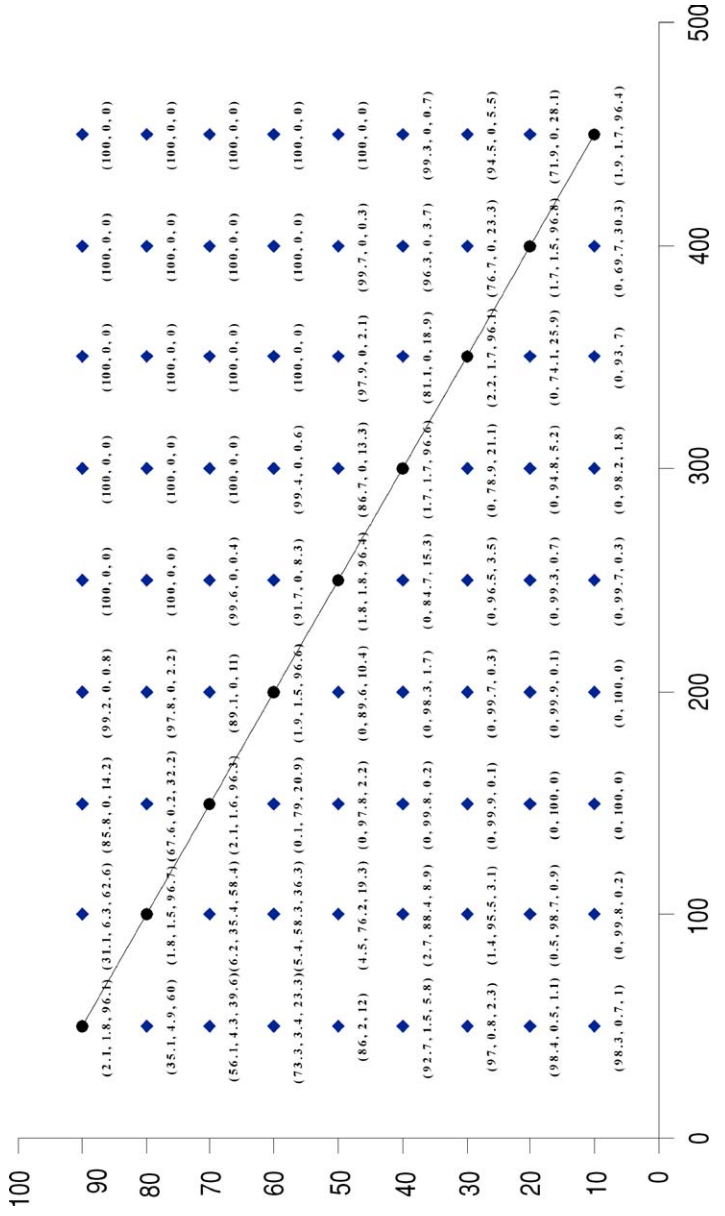
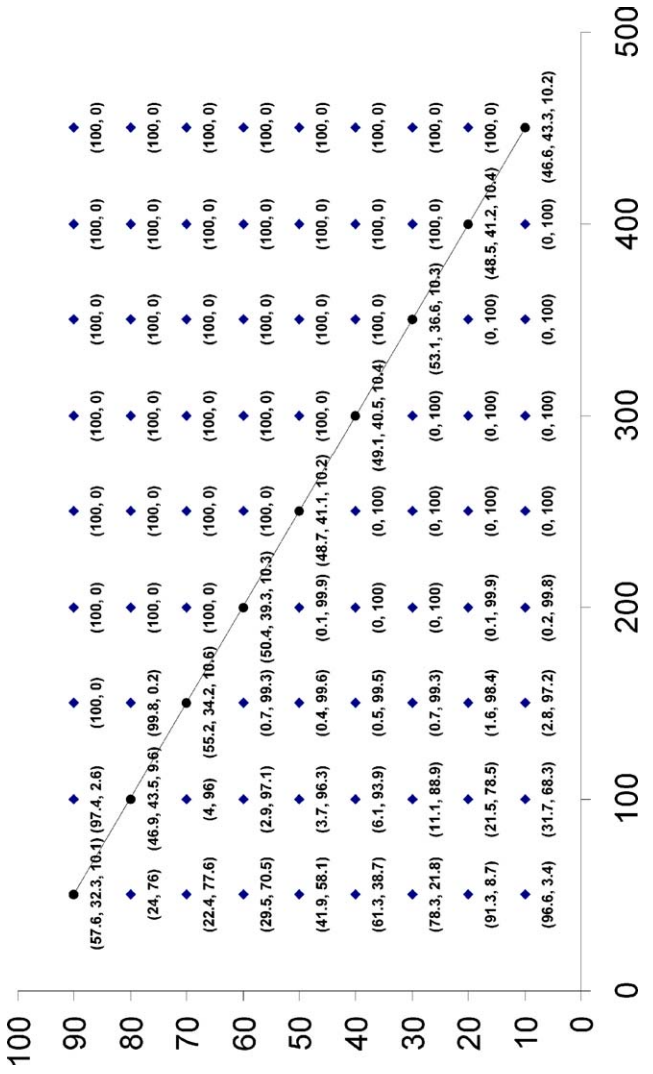


Figure 6. (*Continued*).



(c)

Figure 6. (*Continued*).

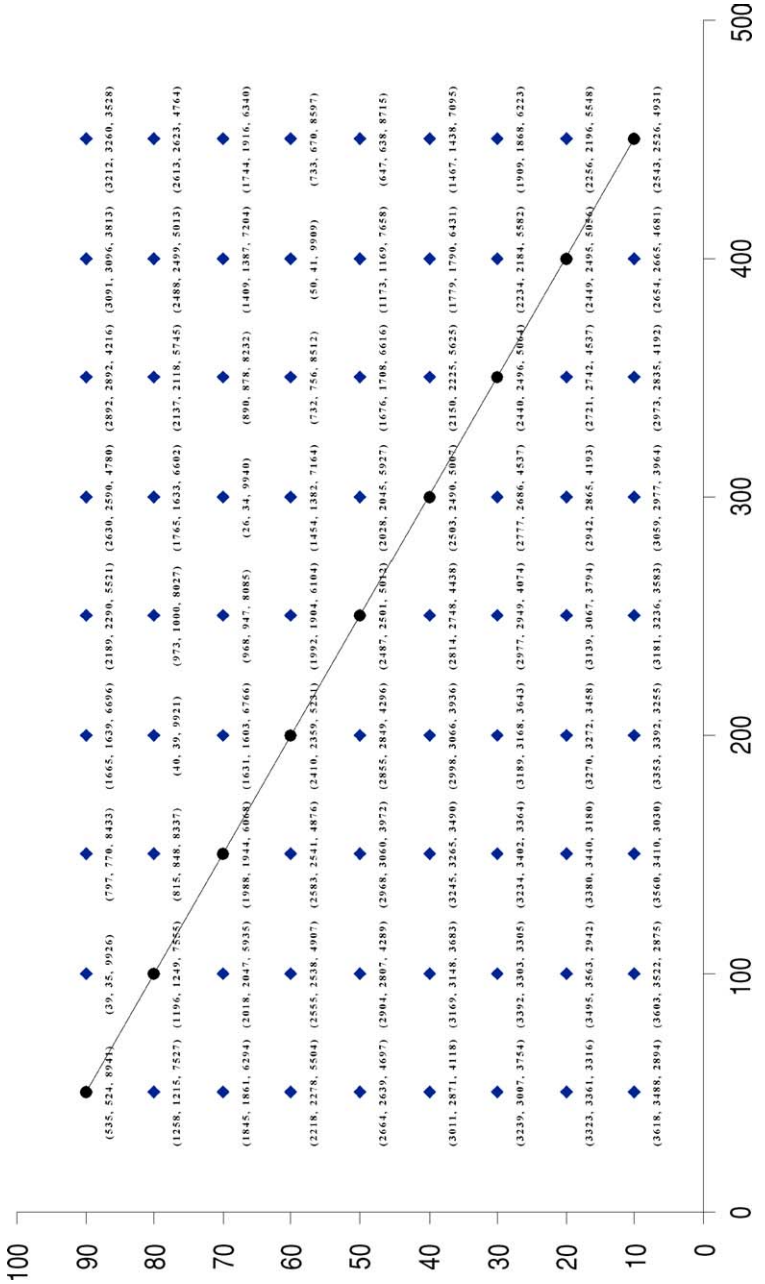
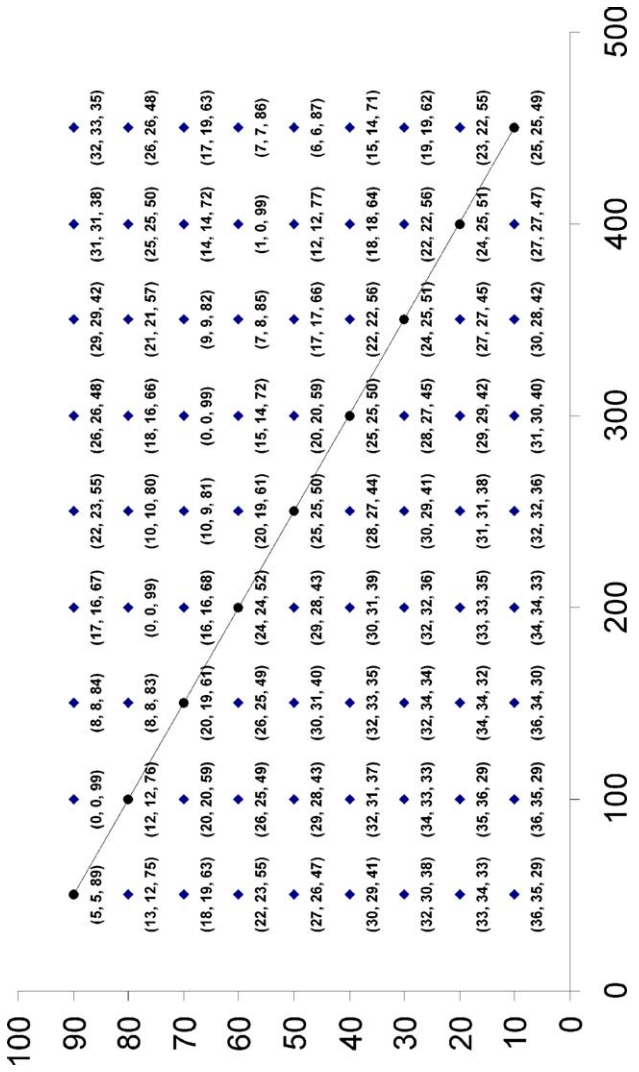
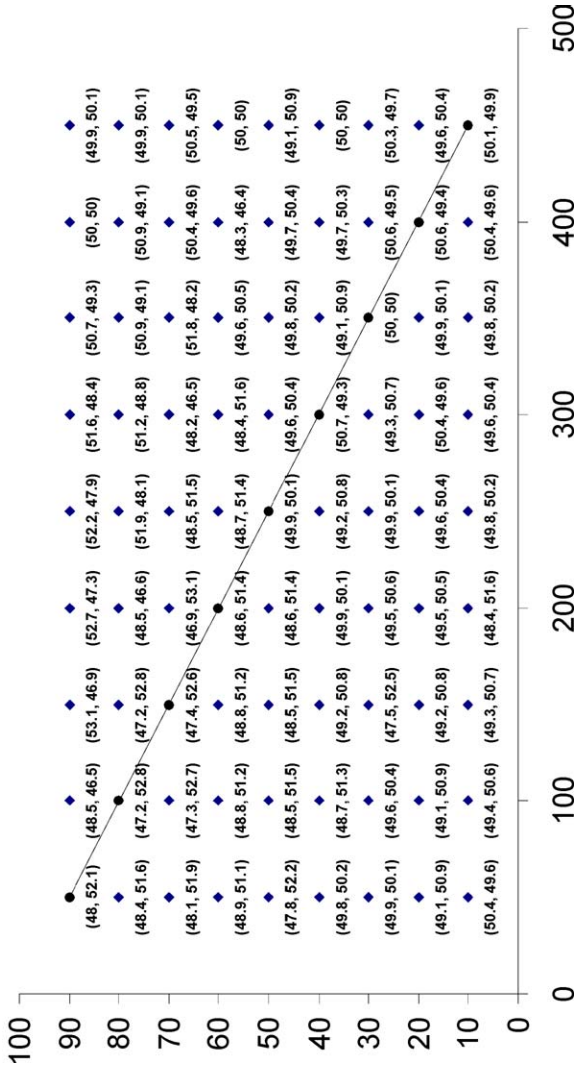


Figure 7. (a): r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0), f_B = single turning point at (x^*, y^*). $P@50$ at (x^*, y^*). $P@50(C)$ vs $P@50(D)$ (<, >, =)—number of cases (total—10, 000). (b): r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0), f_B = single turning point. at (x^*, y^*). $P@50$ at (x^*, y^*), $P@50$(C) vs $P@50$(D). (<, >, =)—percentages (number of cases—10, 000). (c): r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0), f_B = single turning point. at (x^*, y^*). P_{avg} at (x^*, y^*). P_{avg}(C) vs P_{avg}(D). (<, >)—percentages (number of cases—10, 000).

(*Continued on next page.*)


0
100
200
300
400
500
0
10
20
30
40
50
60
70
80
90
100
(5, 5, 89) (0, 0, 99) (8, 8, 84) (17, 16, 67) (22, 23, 55) (26, 26, 48) (29, 29, 42) (31, 31, 38) (32, 33, 35)
(13, 12, 75) (12, 12, 76) (8, 8, 83) (0, 0, 99) (10, 10, 80) (18, 16, 66) (21, 21, 57) (25, 25, 50) (26, 26, 48)
(18, 19, 63) (20, 20, 59) (20, 19, 61) (16, 16, 68) (10, 9, 81) (0, 0, 99) (9, 9, 82) (14, 14, 72) (17, 19, 63)
(22, 23, 55) (26, 25, 49) (26, 25, 49) (24, 24, 52) (20, 19, 61) (15, 14, 72) (7, 8, 85) (1, 0, 99) (7, 7, 86)
(27, 26, 47) (29, 28, 43) (30, 31, 40) (29, 28, 43) (25, 25, 50) (20, 20, 59) (17, 17, 66) (12, 12, 77) (6, 6, 87)
(30, 29, 41) (32, 31, 37) (32, 33, 35) (30, 31, 39) (28, 27, 44) (25, 25, 50) (22, 22, 56) (18, 18, 64) (15, 14, 71)
(32, 30, 38) (34, 33, 33) (32, 34, 34) (32, 32, 36) (30, 29, 41) (28, 27, 45) (24, 25, 51) (22, 22, 56) (19, 19, 62)
(33, 34, 33) (35, 36, 29) (34, 34, 32) (33, 33, 35) (31, 31, 38) (29, 29, 42) (27, 27, 45) (24, 25, 51) (23, 22, 55)
(36, 35, 29) (36, 35, 29) (36, 34, 30) (34, 34, 33) (32, 32, 36) (31, 30, 40) (30, 28, 42) (27, 27, 47) (25, 25, 49)

(b)



(c)

Figure 7. (Continued).

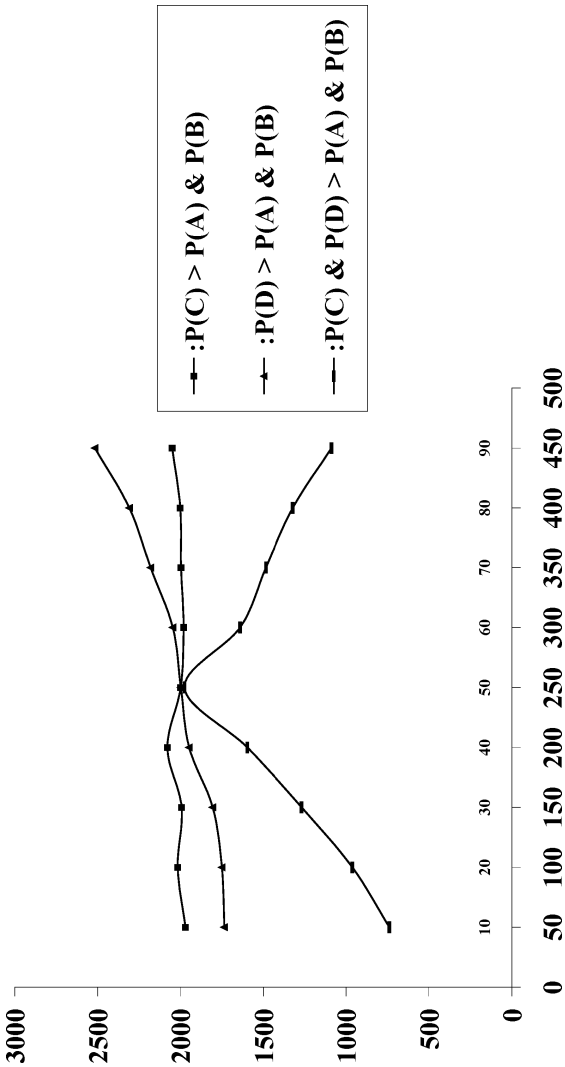
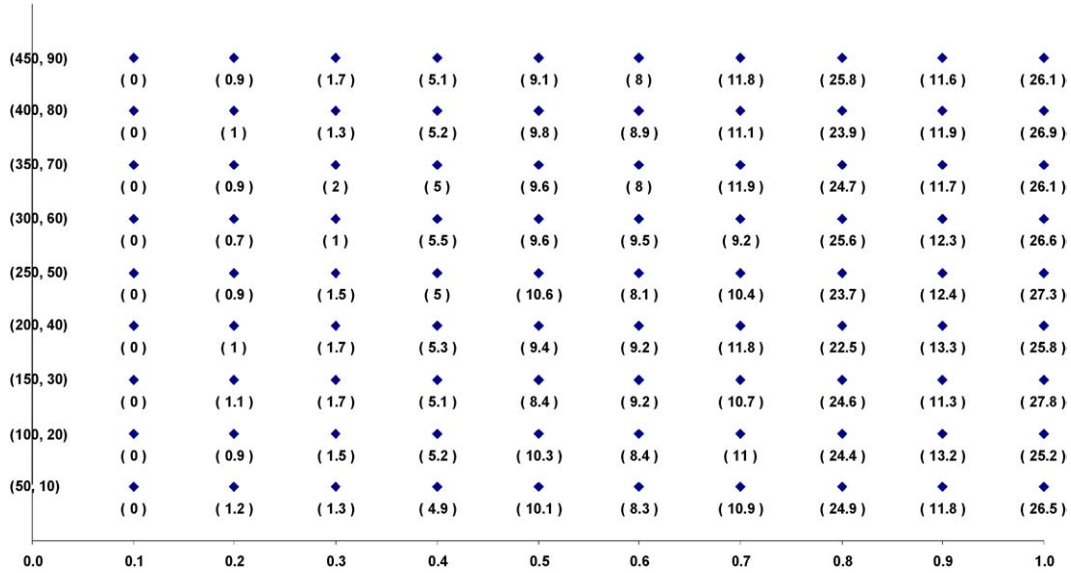


Figure 8. r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0), f_B = single turning point at (x^*, y^*). $P = P_{@50}$ at (x^*, y^*), $x = (x^*, y^*)$ in f_B graph. y = number of cases (total number of cases = 10,000 at each $x = (x^*, y^*)$).

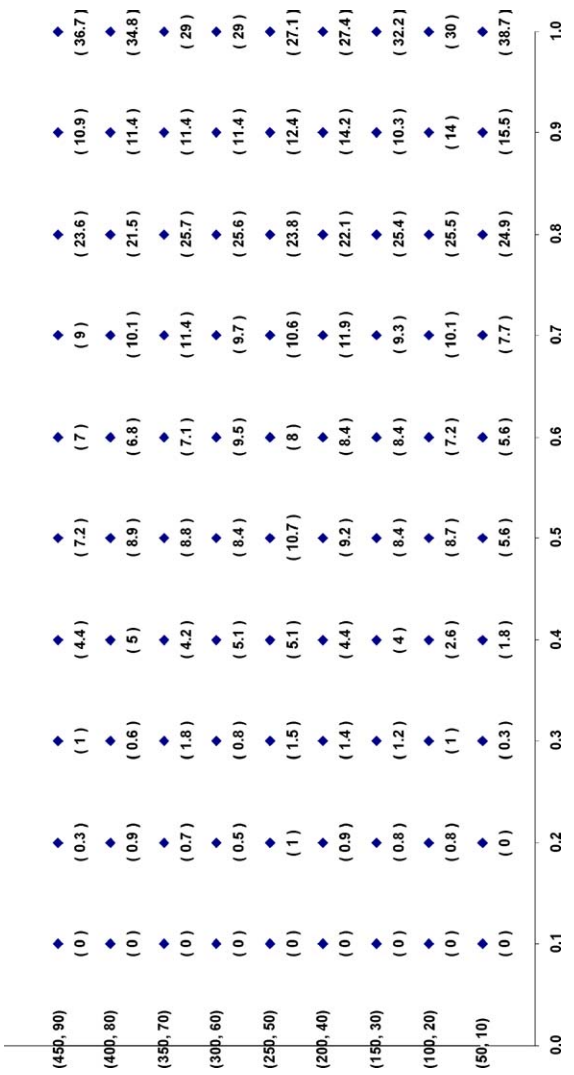


(a)

Figure 9. (a) r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0). $x = P_l/P_h$, $y = t$ and f_B = single turning point $(t, t/5)$. $P = P_{@50}$ at (x, y), $P(C)$ vs $\max\{P(A), P(B)\}$. (>)—percentage $(\sum_{i=1}^{10}(x_i, y) = 100)$, number of cases for each y—10, 000. (b) r_A = random, r_B = random f_A = line between (0,100) and (500,0). $x = P_l/P_h$, $y = t$ and f_B = single turning point $(t, t/5)$. $P = P_{@50}$ at (x, y), $P(D)$ vs $\max\{P(A), P(B)\}$. (>)–percentage ($\sum_{i=1}^{10}(x_i, y) = 100$), number of cases for each y—10, 000. (c) r_A = random, r_B = random. f_A = line between (0, 100) and (500, 0). $x = P_l/P_h$, $y = t$ and f_B = single turning point $(t, t/5)$. $P = P_{@50}$ at (x, y), $\min\{P(C), P(D)\}$ vs max $P(A)$, $P(B$ (>)–percentage $(\sum_{i=1}^{10}(x_i, y) = 100)$, number of cases for each y—10, 000. (d) r_A = random, r_B = random, f_A = line between (0, 100) and (500, 0). $x = P_l/P_h$, $y = t$ and f_B = single turning point $(t, t/5)$. $P = P_{@50}$ at (x, y), (a, b, c), a = percentage of cases where $P(C) > \max\{P(A), P(B)\}$. b = percentage of cases where $P(D) > \max\{P(A), P(B)\}$. c = percentage of cases where $\min\{P(C), P(D)\} > \max\{P(A), P(B)\}$. $\sum_{i=1}^{10}(x_i, y) = 100)$, number of cases for each y—10, 000.

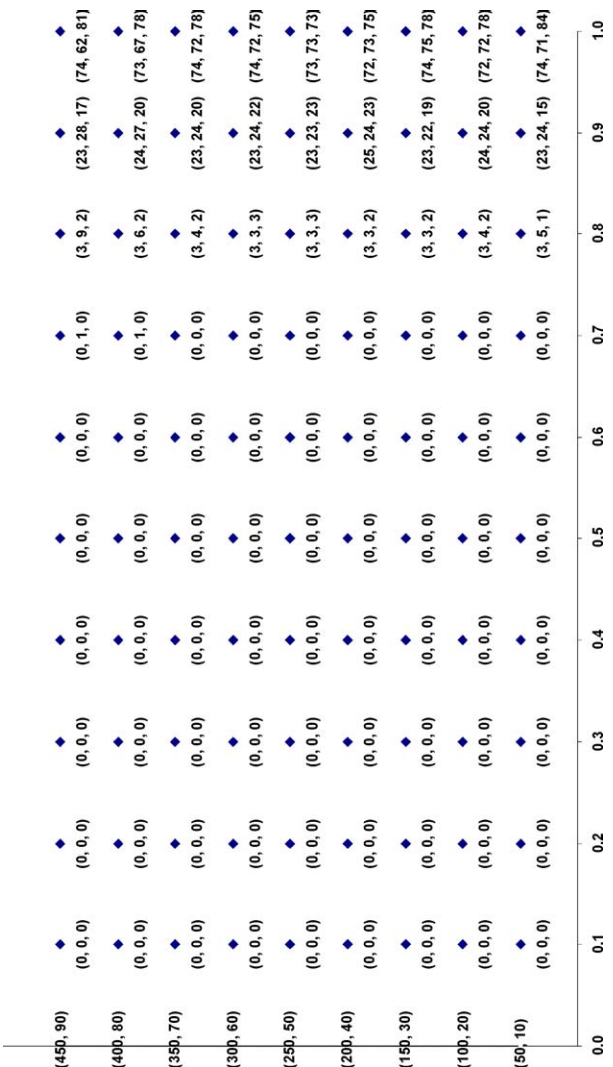
(450, 90)
(400, 80)
(350, 70)
(300, 60)
(250, 50)
(200, 40)
(150, 30)
(100, 20)
(50, 10)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
(0) (1.3) (2.5) (7) (9.9) (8.8) (9.4) (21.7) (12.3) (27.1)
(0) (1.5) (1.6) (5.6) (10.1) (8.5) (10.9) (21.2) (12.1) (28.5)
(0) (0.8) (1.8) (5.3) (9.7) (8.1) (11.5) (24.7) (12.4) (25.7)
(0) (0.7) (0.9) (5.5) (9.2) (9.6) (9.7) (25) (12.3) (27.1)
(0) (0.9) (1.5) (5) (10.6) (8) (10.5) (23.7) (12.3) (27.4)
(0) (1) (1.6) (5.2) (9.7) (8.9) (11.4) (22.5) (13.7) (25.8)
(0) (1.1) (1.3) (4.3) (9.1) (8.3) (10) (25.4) (12.2) (28.3)
(0) (1.2) (1.4) (4.2) (9.7) (7.7) (10.9) (23.3) (14.6) (27)
(0) (0) (0.4) (2) (6.1) (6.3) (9.8) (24.1) (18.7) (32.7)

(b)



(c)

Figure 9. (*Continued*).



(d)

Figure 9. (*Continued*).

References

Aslam JA, Pavlu V and Savell R (2003) A unified model for metasearch, pooling, and system evaluation. In: Proceedings of the Twelfth International Conference on Information and Knowledge Management. New Orleans, LA, pp. 484–491.

Belkin NJ, Cool C, Croft WB and Callan JP (1993) The effect of multiple query representations on information retrieval performance. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, pp. 339–346.

Belkin NJ, Kantor PB, Cool C, and Quatrain R (1994) Combining evidence for information retrieval. In: Harman D (ed.), TREC-2, in: Proceedings of the Second Text Retrieval Conference. Washington, D.C., GPO, pp. 35–44.

Belkin NJ, Kantor PB, Fox EA and Shaw JA (1995) Combining evidence of multiple query representation for information retrieval. Information Processing & Management, 31(3):431–448.

Biggs NL and White T (1979) Permutation Groups and Combinatorial Structures, Cambridge University Press, LMS Lecture Note Series 33.

Chuang H-Y, Liu H, Chen F-A, Kao C-Y and Hsu DF (2004) Combination method in microarray analysis, In: Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'04). IEEE Computer Society Press, pp. 625–630.

Dwork C, Kumar R, Naor M and Sivakumar D (2001) Rank aggregation methods for the web. In: Proceeding of WWW10. Hong Kong, pp. 613–622.

Fagin R, Kumar R and Sivakumar D (2003) Comparing top k-lists. SIAM Journal on Discrete Mathematics. 17:134–160.

Fox EA and Shaw JA (1994) Combination of multiple searches. In: Proceedings of the Second Text Retrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215, pp. 243–252.

Grammatikakis MD, Hsu DF and Kraetzl M (2001) Parallel System Interconnections and Communications. CRC Press.

Heydemann MC (1997) Cayley graphs and interconnection networks. In Hahn G. and Sabidussi G. (eds.), Graph Symmetry. Kluwer Academic Publishers, pp. 161–224.

Hsu DF, Lyons DM, Usandivaras C and Montero F (2003) RAF: A dynamic and efficient approach to fusion for multiple target tracking in CCTV surveillance. In: Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Inteligent Systems (MFI). IEEE Computer Society Press, pp. 222–228.

Hsu DF and Palumbo A (2004) A study of data Fusion in Cayley Graphs $G(S_n, P_n)$. In: Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN'04). IEEE Computer Society Press, pp. 557–562.

Hsu DF, Shapiro J and Taksa I (2002) Methods of data fusion in information retrieval: Rank vs. score combination, DIMACS Technical Report 2002–58, pp. 1–47.

Ibraev U, Ng KB and Kantor PB (2001) Counter intuitive cases of data fusion in information retrieval. Rutgers University Technical Report.

Kantor PB (1998) Semantic dimension: On the effectiveness of naive data fusion methods in certain learning and detection problems. In: Fifth International Symposium on Artificial Intelligence and Mathematics. Ft. Lauderdale, FL.

Lee JH (1997) Analyses of multiple evidence combination. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Philadelphia, PA, pp. 267–276.

Lyons DM, Hsu DF, Usandivaras C and Montero F (2003) Experimental results from using rank and fuse approach for multi-target tracking in CCTV surveillance. In: Proceedings of IEEE International Conference on AVSS. IEEE Computer Society Press, pp. 345–351.

Marden JI (1995) Analyzing and modeling rank data. Monographs on Statistics and Applied Probability No. 64, Chapman & Hall.

Ng KB and Kantor PB (1998) An investigation of the preconditions for effective data fusion in information retrieval: A pilot study. In: Proceedings of the 61st Annual Meeting of the American Society for Information Science, pp. 166–178.

Ng KB and Kantor PB (2000) Predicting the effectiveness of naïve data fusion on the basis of system characteristics, Journal of the American Society for Information Science, 51(13):1177–1189.

Pfeifer U, Poersch T and Fuhr N (1996) Retrieval effectiveness of proper name search methods. Information Processing and Management, 32(6):667–679.

van Rijsbergen CJ (1986) A new theoretical framework of information retrieval. In: Proceedings of the 9th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Pisa, Italy, pp. 194–200.

Saracevic T and Kantor PB (1988) A study of information seeking and retrieving. III Searchers, searches, overlap. Journal of the ASIS, 39:197–216.

Varshney PK (ed.) (1997) In: Proceedings of the IEEE. Special issue on data fusion 85(1) pp. 3–183.

Vogt CC and Cottrell GW (1999) Fusion via a linear combination of scores. Information Retrieval, 1(3):151–173.

Xu L, Krzyzak A and Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics, 22(3):418—435.