Analyses of Multiple Evidence Combination

Joon Ho Lee*
Korean Research and Development Information Center
P.O. Box 122, Yusong, Taejon 305-600, Korea
joonho@kordic.re.kr

Abstract

It has been known that different representations of a query retrieve different sets of documents. Recent work suggests that significant improvements in retrieval performance can be achieved by combining multiple representations of an information need. However, little effort has been made to understand the reason why combining multiple sources of evidence improves retrieval effectiveness. In this paper we analyze why improvements can be achieved with evidence combination, and investigate how evidence should be combined. We describe a rationale for multiple evidence combination, and propose a combining method whose properties coincide with the rationale. We also investigate the effect of using rank instead of similarity on retrieval effectiveness.

1 Introduction

A variety of representation techniques for queries and documents have been proposed in the information retrieval (IR) literature, and many corresponding retrieval techniques have also been developed to get higher retrieval effectiveness. Recent research shows that retrieval effectiveness can be improved by using multiple query or document representations, or multiple retrieval techniques, and combining the retrieval results, in contrast to using just a single representation or a single retrieval technique. This general area has been discussed in the literature under the name of "data fusion".

Turtle & Croft [11] developed an inference network-based retrieval model, which can combine different document representations and different versions of a query in a consistent probabilistic framework. They implemented the INQUERY retrieval system based on the model, and demonstrated that multiple evidence increases retrieval effectiveness in some circumstances. Fox & Shaw [4] and Bartell, et al. [1] have worked on various methods for combining multiple retrieval runs, and have obtained improvements over any single retrieval run. Belkin, et al. [2] showed that progressive combination of different Boolean query formulations could lead to progressive improvements of retrieval effectiveness. Lee [7] described how different properties of weighting schemes may retrieve different types of documents, and showed that significant improvements could be obtained by combining the retrieval results from weighting schemes with different properties.

The research results described above show that combining multiple types of evidence can improve the effectiveness of information retrieval. However, little effort has been made to figure out the reason why combining evidence improves retrieval effectiveness. Consequently, the particular combining functions used in data fusion have received little justification. In this paper we provide experimental results supporting that data fusion improves retrieval effectiveness, but the primary aim of our study is to understand why improvements can be achieved by multiple evidence combination, and also to investigate how evidence should be combined.

Belkin, et al. [2] argued that different runs retrieve different sets of relevant documents and also retrieve different sets of nonrelevant documents. We analyze research results obtained in the data fusion literature, and give a new rationale for evidence combination that is a little different from the previous one. That is, we show that different runs retrieve similar sets of relevant documents but retrieve different sets of nonrelevant documents. We evaluate a variety of combining methods, and show that the function called CombMNZ provides better retrieval effectiveness than the others. We describe the relationship between our rationale and the properties of the CombMNZ function. We also investigate the effect of using rank instead of similarity on retrieval effectiveness, and show that using rank works better than using similarity in some circumstances.

The remainder of this paper is organized as follows. In Section 2 we analyze research results performed in the data fusion literature, and give a new rationale for multiple evidence combination. In Section 3 we evaluate a variety of combining methods, and explain the relationship between our rationale and the best combining method. Section 4 describes the effect of using rank instead of similarity on retrieval effectiveness. Finally, concluding remarks are given in Section 5.

2 Rationales for Multiple Evidence Combination

Belkin, et al. [2] combined different Boolean query formulations, and showed that the combination could lead to improvements of retrieval effectiveness. They also gave a rationale for data fusion, which is quoted below.

*This work was done while visiting the NSF Center for Intelligent Information Retrieval at the University of Massachusetts, Amherst, MA during April to June in 1996.

SIGIR 97 Philadelphia PA, USA

> Different representations of the same query, or of the documents in the database, or different retrieval techniques for the same query, retrieve different sets of documents (both relevant and nonrelevant).

The above rationale derives from two earlier research results as follows: First, McGill, Koll & Norreault [9] found that there was surprisingly little overlap between document sets for the same information need when documents were retrieved by different users or by the same user using controlled versus free-text vocabularies. Second, Katzer, et al. [6] considered the effect of different document representations (e.g., title, abstract) on retrieval effectiveness rather than different query representations. They discovered that the various document representations gave similar retrieval effectiveness, but retrieved quite different sets of documents.

Other research results suggest that this rationale for multiple evidence combination should be investigated in more detail. Turtle & Croft [11] evaluated both probabilistic and Boolean versions of queries, and combined the results. Combining queries resulted in significant performance improvements for the CACM and CISI collections. They also gave an interesting analysis, which is quoted below.

> We originally thought that at least part of the performance improvements arose because the two query types were retrieving different relevant documents, so that the combined set contained more relevant documents than retrieved by the separate queries. This is not, however, the case. The documents retrieved by the Boolean queries are a subset of those retrieved by the corresponding probabilistic query.

The above analysis suggests that improvements could be achieved by combining two different runs even if they retrieve similar sets of documents, and should lead to a modification of the rationale given by Belkin, et al.

Saracevic & Kantor [10] asked different experts to construct Boolean queries based on the same description of an information problem in operational online information retrieval systems. Like McGill, Koll & Norreault and Katzer, et al., they found that different query formulations generated different documents. However, they noticed that the odds of a document being judged relevant increased monotonically with the number of retrieved sets in which the document appeared. These results could lead to a new rationale for evidence combination: *different runs might retrieve similar sets of relevant documents but retrieve different sets of nonrelevant documents.*

In order to justify our new rationale, we compute the overlap coefficients called $R_{overlap}$ and $N_{overlap}$ that show the degree of overlap among relevant documents and nonrelevant documents in two retrieval results. The coefficients $R_{overlap}$ and $N_{overlap}$ are defined for two runs $run1$ and $run2$ as follows:

$$R_{overlap} = \frac{R_{common} \times 2}{R_1 + R_2}$$

$$N_{overlap} = \frac{N_{common} \times 2}{N_1 + N_2}$$

R_{common} number of common relevant documents

R_1 number of relevant documents in $run1$

R_2 number of relevant documents in $run2$

N_{common} number of common nonrelevant documents

N_1 number of nonrelevant documents in $run1$

N_2 number of nonrelevant documents in $run2$

$R_{overlap}$ is 1 if $run1$ and $runs2$ retrieve identical sets of relevant documents, and 0 if they do not retrieve any common relevant document. $N_{overlap}$ is 1 if $run1$ and $runs2$ retrieve identical sets of nonrelevant documents, and 0 if they do not retrieve any common nonrelevant document.

Data fusion is often performed within a single retrieval system in that one retrieval system generates and combines multiple types of evidence to improve retrieval effectiveness. In this paper, however, we will investigate retrieval results produced by quite different retrieval systems rather than one retrieval system. Many systems participate in the TREC conference [5], in which the systems retrieve top-ranked documents for the given document and query sets. Since the top-ranked documents are generated using the same document and query sets, combining the different results can be considered a kind of data fusion. In the remainder of this paper, we will exploit several retrieval results submitted to the TREC3 ad-hoc track.

We selected six retrieval results from the TREC3 ad-hoc track, namely ***westp1***, ***pircs1***, ***vtc5s2***, ***brkly6***, ***eth001*** and ***nyuir1***. We calculated the overlap coefficients for pairwise combinations of the six retrieval results. Table 1 shows the coefficients $R_{overlap}$ and $N_{overlap}$ for each combination. We can easily see that the degree of overlap among relevant documents, i.e. $R_{overlap}$ is much greater than the degree of overlap among nonrelevant documents, i.e. $N_{overlap}$, which will be called the *unequal overlap property*. This property indicates that different runs retrieve similar sets of relevant documents but retrieve different sets of nonrelevant documents. The unequal overlap property also coincides with Saracevic & Kantor's results that *the more runs a document is retrieved by, the higher the rank that should be assigned to the document.*

3 Combining Methods

Since different retrieval results can generate quite different ranges of similarity values, a normalization method should be applied to each retrieval result. For TREC topic 151, for instance, the six different retrieval results from the TREC3 ad-hoc track give the maximum and minimum similarity values shown in Table 2. Normalization controls the ranges of similarity values that retrieval systems generate. Hence, in order to align both the lower bounds of similarity values and the upper, we normalize each similarity value by the maximum and minimum actually seen in a retrieval result as follows:

$$normalized\_sim = \frac{unnormalized\_sim - min\_sim}{max\_sim - min\_sim}$$

Fox & Shaw [4] tested several functions for combining multiple evidence, namely CombMIN, CombMAX, CombSUM, CombANZ and CombMNZ, which are shown in Table 3. They performed five different retrieval runs, and combined the retrieval results. Two types of queries were used, P-norm extended Boolean queries [8] and natural language vector queries. A single set of P-norm queries was created, but it was interpreted multiple times with different operator weights (P-values) of 1.0, 1.5 and 2.0; these three runs are designated $Pn1.0$, $Pn1.5$, and $Pn2.0$. Two sets of vector queries were automatically constructed directly from TREC

Table 1: Degree of overlap among relevant and nonrelevant documents (six retrieval results are selected from the TREC3 ad-hoc track; numbers, i.e. num_of_common_rel_docs, et al. are summed up for 50 queries)

		westp1	*pircs1*	*vtc5s2*	*brkly6*	*eth001*
pircs1	$R_{overlap}$	0.7970				
	$N_{overlap}$	0.3620				
vtc5s2	$R_{overlap}$	0.7712	0.7562			
	$N_{overlap}$	0.3009	0.3035			
brkly6	$R_{overlap}$	0.7846	0.7813	0.7846		
	$N_{overlap}$	0.3522	0.3649	0.3272		
eth001	$R_{overlap}$	0.7706	0.7927	0.7686	0.8253	
	$N_{overlap}$	0.3260	0.3869	0.2936	0.4179	
nyuir1	$R_{overlap}$	0.7902	0.8210	0.7457	0.7562	0.7882
	$N_{overlap}$	0.3517	0.4360	0.3303	0.3238	0.4009

Table 2: Maximum and minimum similarity values generated with respect to TREC topic 151 (six retrieval results are selected from the TREC3 ad-hoc track)

	maximum_similarity	minimum_similarity
westp1	0.7533	0.6567
pircs1	6.1525	2.0258
vtc5s2	1.8289	0.6860
brkly6	0.4682	0.1415
eth001	0.3780	0.0903
nyuir1	28643	6326

topic descriptions. One of the sets included the Narrative section of the topic descriptions; this set is referred to as the long vector (LV) query set while the other is referred to as the short vector (SV) query set.

Table 4 shows the effectiveness of the five individual runs and the combination runs. As shown in the table, the summation function, which sums up the set of similarity values, works better in the TREC subcollections such as AP-1, WSJ-1, AP-2 and WSJ-2 [3]. Our analyses suggested that the more runs a document is retrieved by, the higher the rank that combining functions should assign to the document. Part of Fox & Shaw's result agrees with our analyses, in that CombSUM, which favors the documents retrieved by more runs, provides better retrieval effectiveness than CombMAX, CombMIN and CombANZ. On the other hand, part of the result does not seem to coincide with our analyses, in that CombMNZ should be better than CombSUM or at least not much worse because it has the property of favoring the documents retrieved by more runs.

One important difference between Fox & Shaw's and our experiments lies in when the combination is performed. Fox & Shaw combined multiple evidence at retrieval time, and did not apply any normalization method to individual runs. However, we combine the results retrieved by multiple systems, and normalize each similarity value by the maximum and minimum similarity values in a retrieval result.

We have applied the functions used by Fox & Shaw to pairwise combinations of the six runs from the TREC3 ad-hoc track to see how much these different approaches affect retrieval effectiveness. Table 5 shows the results. In this experiment, CombMNZ works slightly better than CombSUM. We have also combined not only all pairwise combinations (called 2-way from now on) but also all 3-way, 4-way, 5-way combinations of the six runs, and the combination of all six runs. Table 6 shows the average precision for the average of all combined runs in each level of combination. The table shows that CombMNZ gives still better retrieval effectiveness than CombSUM.

The CombSUM and CombMNZ functions can be generalized to the following function, which will be designated as CombGMNZ.

$$CombGMNZ = CombSUM \times num\_of\_nonzero\_sims^{\gamma}$$

where $\gamma \geq 0$. CombGMNZ is equivalent to CombSUM if γ is equal to zero, and CombGMNZ is equivalent to CombMNZ if γ is equal to one. The parameter γ is concerned with how much higher weights are given to the documents retrieved by more retrieval runs. We have applied the CombGMNZ function to pairwise combinations of the six retrieval results from the TREC3 ad-hoc track by changing the value of the parameter γ. Table 7 shows that the CombGMNZ function provides the best results when the value of the parameter γ is equal to one.

We have also investigated for TREC topic 151 the number of common documents retrieved by two runs of each pairwise combination, and the rank of the top-ranked document retrieved by only one of two runs. Table 8 shows that the rank of the top-ranked document retrieved by only one run is increased with the value of the parameter γ. We also see that even if we assign the value ten to the parameter γ to make all common documents ranked higher than any document that is not common, the effectiveness of the function is still slightly better than that of CombSUM.

Since the number of retrieved documents could has influence on the effectiveness of the combining functions, some might think that CombSUM works better than CombMNZ if smaller number of documents is retrieved than 1000. We have applied the functions CombSUM and CombMNZ to the top-ranked 50, 100, 200, 400, 600 and 800 documents. Table 9 shows that CombMNZ always works slightly better than CombSUM regardless of the number of retrieved documents.

4 Using Rank vs Similarity

In the data fusion literature similarity is more often exploited to combine multiple evidence than rank values. We

Table 3: Combining functions proposed by Fox & Shaw

CombMIN	minimum of individual similarities
CombMAX	maximum of individual similarities
CombSUM	summation of individual similarities
CombANZ	CombSUM ÷ number of nonzero similarities
CombMNZ	CombSUM × number of nonzero similarities

Table 4: Average precision for the combining functions without similarity normalization

Run	AP-1	WSJ-1	AP-2	WSJ-2	average
*Pn*1.0	0.2810	0.2941	0.3004	0.2206	0.2740
*Pn*1.5	0.3122	0.3199	0.3332	0.2327	0.2995
*Pn*2.0	0.3027	0.3217	0.3300	0.2325	0.2967
SV	0.2387	0.2203	0.2543	0.1503	0.2159
LV	0.2435	0.2414	0.2664	0.1633	0.2287
CombMIN	0.2863	0.1924	0.3047	0.1308	0.2286
CombMAX	0.2856	0.3205	0.3337	0.2343	0.2935
CombSUM	0.3493	0.3605	0.3748	0.2752	0.3340
CombANZ	0.3493	0.3367	0.3748	0.2465	0.3268
CombMNZ	0.3059	0.3368	0.3516	0.2467	0.3103

suggest two reasons why only similarity values have been taken into consideration. First, people seem to believe that using similarity gives more benefits than using rank in terms of retrieval effectiveness. Second, the ranks of documents in individual runs have not been available at "fusion" time since the multiple evidence combination has been performed at retrieval time (i.e., before all documents are ranked).

One of reasons why we investigate the use of rank values is that using similarity might give less retrieval effectiveness than using rank in certain cases. This is because using similarity has the effect of weighting individual runs without considering their overall performance, which will be called the *independent weighting effect*. For example, we plotted the normalized similarity values of the documents retrieved by the six runs from the TREC3 ad-hoc track with respect to TREC topic 151. Figure 1 shows that the normalized similarity value of *pircs1* is less than that of *brkly6* for each rank, which results in favoring *brkly6* in the combination of *pircs1* and *brkly6*. If the run *pircs1* had better retrieval effectiveness than the run *brkly6* for TREC topic 151, the independent weighting effect would decrease retrieval effectiveness in their combination.

We have combined the results retrieved by multiple systems after retrieval time, and thus the ranks of documents are available. In what follows, we investigate the effect of using rank instead of similarity in combining multiple evidence. First we apply the following function called Rank_Sim to the rank of a document, and used the resulting value as the similarity value of the document.

$$Rank\_Sim(rank) = 1 - \frac{rank - 1}{num\_of\_retrieved\_docs}$$

For example, suppose an individual run retrieves top-ranked 1000 documents. Given a document ranked at 10, the similarity value of the document is equal to 0.991. We evaluated the effectiveness of the combination runs in which the similarity of a document is transformed from the rank of the document. We applied the CombMNZ function to pairwise combinations of the six runs from the TREC3 ad-hoc track. Table 10 shows that similarity values provide slightly better retrieval effectiveness than rank values.

The results given in Table 10 are a little different from our hypothesis that using similarity values might be worse than using rank values in terms of retrieval effectiveness in certain cases. This seems to be caused by the the following reasons:

- Suppose that *run1* provides better retrieval effectiveness than *run2* on average while *run2* is favored over *run1* in their combination due to the independent weighting effect. One might think that combining similarity values would hurt retrieval effectiveness because the worse run *run2* is favored over the better run *run1*. However, we should notice that the worse run *run2* gives better retrieval effectiveness than the better run *run1* for some queries. For instance, Table 11 shows average precision for TREC topics 151-200. we can see that *nyuir1* provides better retrieval effectiveness than *westp1* for 18 TREC topics even though *westp1* is significantly better than *nyuir1* on average. Therefore, the independent weighting effect produces some gains in terms of retrieval effectiveness for some queries, which might be able to offset the losses resulting from the independent weighting effect.

- We have explained that the more runs a document is retrieved by, the higher the rank that should be assigned to the document in order to get higher retrieval effectiveness. We have also shown that the function CombMNZ favors the documents retrieved by more runs so that it provides better retrieval effectiveness than the others. The independent weighting effect does not affect the property of CombMNZ that favors documents retrieved by more runs. This might be one of reasons why using similarity does not decrease retrieval effectiveness even though it causes the independent weighting effect.

Table 5: Average precision for the combining functions with similarity normalization (functions are applied to pairwise combinations of the six retrieval results from the TREC3 ad-hoc track; averages over 50 queries)

	CombMIN	CombMAX	CombSUM	CombANZ	CombMNZ
westp1 & *pircs1*	0.2780	0.3377	0.3586	0.3280	0.3604
westp1 & *vtc5s2*	0.2524	0.3355	0.3690	0.3100	0.3737
westp1 & *brkly6*	0.2641	0.3238	0.3490	0.3111	0.3525
westp1 & *eth001*	0.2560	0.3258	0.3502	0.3112	0.3522
westp1 & *nyuir1*	0.2566	0.3205	0.3505	0.3054	0.3559
pircs1 & *vtc5s2*	0.2469	0.3221	0.3463	0.2994	0.3520
pircs1 & *brkly6*	0.2595	0.3083	0.3275	0.2979	0.3305
pircs1 & *eth001*	0.2637	0.3115	0.3287	0.3045	0.3293
pircs1 & *nyuir1*	0.2663	0.3062	0.3259	0.3014	0.3273
vtc5s2 & *brkly6*	0.2456	0.3106	0.3390	0.2911	0.3431
vtc5s2 & *eth001*	0.2295	0.3108	0.3468	0.2876	0.3543
vtc5s2 & *nyuir1*	0.2532	0.3150	0.3341	0.3027	0.3334
brkly6 & *eth001*	0.2554	0.2900	0.3142	0.2876	0.3170
brkly6 & *nyuir1*	0.2378	0.3011	0.3354	0.2834	0.3402
eth001 & *nyuir1*	0.2564	0.2877	0.3214	0.2963	0.3230
average	0.2548	0.3144	0.3398	0.3012	0.3430

Table 6: Average precision for the combining functions with similarity normalization (functions are applied to 3- to 6-way combinations of the six retrieval results from the TREC3 ad-hoc track; averages of all combined runs in each level of combination)

	1-way[1]	2-way[2]	3-way	4-way	5-way	6-way
CombMIN	0.2884	0.2548	0.2237	0.2012	0.1849	0.1720
CombMAX	0.2884	0.3144	0.3273	0.3350	0.3404	0.3460
CombSUM	0.2884	0.3398	0.3646	0.3797	0.3899	0.3972
CombANZ	0.2884	0.3012	0.3058	0.3088	0.3115	0.3134
CombMNZ	0.2884	0.3430	0.3685	0.3835	0.3927	0.3991

[1] average of average precisions for the six retrieval results

[2] repeated from Table 5

Table 7: Average precision for the function CombGMNZ (six runs are selected from the TREC3 ad-hoc track; averages over 50 queries)

	$\gamma = 0$ CombSUM	$\gamma = 0.5$	$\gamma = 1$ CombMNZ	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$
westp1 & *pircs1*	0.3585	0.3603	0.3604	0.3599	0.3596	0.3596
westp1 & *vtc5s2*	0.3690	0.3731	0.3737	0.3728	0.3718	0.3718
westp1 & *brkly6*	0.3490	0.3519	0.3525	0.3520	0.3512	0.3511
westp1 & *eth001*	0.3502	0.3523	0.3522	0.3510	0.3501	0.3501
westp1 & *nyuir1*	0.3505	0.3548	0.3559	0.3557	0.3551	0.3550
pircs1 & *vtc5s2*	0.3463	0.3509	0.3520	0.3520	0.3515	0.3515
pircs1 & *brkly6*	0.3275	0.3302	0.3305	0.3300	0.3297	0.3297
pircs1 & *eth001*	0.3287	0.3295	0.3293	0.3285	0.3278	0.3278
pircs1 & *nyuir1*	0.3259	0.3272	0.3273	0.3264	0.3257	0.3257
vtc5s2 & *brkly6*	0.3390	0.3427	0.3431	0.3423	0.3418	0.3418
vtc5s2 & *eth001*	0.3468	0.3526	0.3543	0.3543	0.3537	0.3537
vtc5s2 & *nyuir1*	0.3341	0.3345	0.3334	0.3318	0.3308	0.3307
brkly6 & *eth001*	0.3142	0.3164	0.3170	0.3172	0.3173	0.3173
brkly6 & *nyuir1*	0.3354	0.3393	0.3402	0.3401	0.3395	0.3395
eth001 & *nyuir1*	0.3214	0.3230	0.3230	0.3224	0.3217	0.3217
average	0.3398	0.3426	0.3430	0.3424	0.3418	0.3418

Table 8: Rank of the top-ranked document retrieved by only one of two runs with respect to TREC topic 151 (six runs are selected from the TREC3 ad-hoc track)

	$\gamma = 0$ CombSUM	$\gamma = 0.5$	$\gamma = 1$ CombMNZ	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$ num of common docs
westp1 & *pircs1*	208	251	284	343	380	381
westp1 & *vtc5s2*	146	193	230	292	328	331
westp1 & *brkly6*	186	246	290	360	400	400
westp1 & *eth001*	127	178	211	286	337	337
westp1 & *nyuir1*	216	257	302	398	462	463
pircs1 & *vtc5s2*	193	303	410	523	581	581
pircs1 & *brkly6*	221	330	448	596	665	665
pircs1 & *eth001*	105	186	304	521	656	661
pircs1 & *nyuir1*	279	385	515	661	735	739
vtc5s2 & *brkly6*	179	267	369	524	618	619
vtc5s2 & *eth001*	213	328	428	542	611	612
vtc5s2 & *nyuir1*	215	324	409	538	611	612
brkly6 & *eth001*	225	352	464	636	731	731
brkly6 & *nyuir1*	157	253	360	552	712	712
eth001 & *nyuir1*	209	377	489	647	742	743
average	191.9	282.0	367.5	494.6	571.3	572.5

Table 9: Average precision for CombMNZ and CombSUM when the functions are applied to top-ranked N documents (six runs are selected from the TREC3 ad-hoc track; averages over 50 queries)

		Top-50	Top-100	Top-200	Top-400	Top-600	Top-800
westp1 & *pircs1*	CombMNZ	0.1511	0.2135	0.2686	0.3207	0.3428	0.3530
	CombSUM	0.1506	0.2118	0.2678	0.3187	0.3410	0.3517
westp1 & *vtc5s2*	CombMNZ	0.1471	0.2103	0.2721	0.3292	0.3549	0.3666
	CombSUM	0.1443	0.2053	0.2660	0.3229	0.3492	0.3616
westp1 & *brkly6*	CombMNZ	0.1463	0.2069	0.2626	0.3098	0.3338	0.3453
	CombSUM	0.1438	0.2030	0.2590	0.3063	0.3299	0.3420
westp1 & *eth001*	CombMNZ	0.1450	0.2014	0.2572	0.3085	0.3333	0.3447
	CombSUM	0.1437	0.1987	0.2539	0.3062	0.3303	0.3423
westp1 & *nyuir1*	CombMNZ	0.1448	0.2038	0.2627	0.3140	0.3363	0.3482
	CombSUM	0.1395	0.1987	0.2563	0.3088	0.3310	0.3428
pircs1 & *vtc5s2*	CombMNZ	0.1403	0.2005	0.2573	0.3091	0.3317	0.3445
	CombSUM	0.1379	0.1964	0.2522	0.3030	0.3257	0.3382
pircs1 & *brkly6*	CombMNZ	0.1380	0.1959	0.2488	0.2919	0.3113	0.3227
	CombSUM	0.1368	0.1932	0.2452	0.2886	0.3086	0.3201
pircs1 & *eth001*	CombMNZ	0.1398	0.1929	0.2449	0.2899	0.3098	0.3213
	CombSUM	0.1385	0.1905	0.2422	0.2886	0.3089	0.3206
pircs1 & *nyuir1*	CombMNZ	0.1385	0.1909	0.2444	0.2908	0.3092	0.3201
	CombSUM	0.1330	0.1877	0.2417	0.2882	0.3075	0.3181
vtc5s2 & *brkly6*	CombMNZ	0.1355	0.1949	0.2521	0.3020	0.3239	0.3357
	CombSUM	0.1329	0.1911	0.2468	0.2964	0.3187	0.3311
vtc5s2 & *eth001*	CombMNZ	0.1365	0.1937	0.2524	0.3093	0.3331	0.3457
	CombSUM	0.1329	0.1884	0.2444	0.3002	0.3242	0.3376
vtc5s2 & *nyuir1*	CombMNZ	0.1296	0.1868	0.2439	0.2965	0.3169	0.3274
	CombSUM	0.1264	0.1831	0.2407	0.2933	0.3153	0.3269
brkly6 & *eth001*	CombMNZ	0.1300	0.1841	0.2354	0.2786	0.2979	0.3091
	CombSUM	0.1279	0.1801	0.2314	0.2743	0.2943	0.3062
brkly6 & *nyuir1*	CombMNZ	0.1358	0.1931	0.2479	0.2982	0.3189	0.3319
	CombSUM	0.1310	0.1869	0.2404	0.2908	0.3129	0.3267
eth001 & *nyuir1*	CombMNZ	0.1311	0.1832	0.2353	0.2843	0.3045	0.3157
	CombSUM	0.1276	0.1786	0.2301	0.2795	0.3008	0.3131
average	CombMNZ	0.1393	0.1968	0.2524	0.3022	0.3239	0.3355
	CombSUM	0.1365	0.1929	0.2479	0.2977	0.3199	0.3319

Table 10: Comparison of rank combination and similarity combination (six runs are selected from the TREC3 ad-hoc track; averages over 50 queries)

		westp1	*pircs1*	*vtc5s2*	*brkly6*	*eth001*
pircs1	rank	0.3525				
	similarity	0.3604				
vtc5s2	rank	0.3654	0.3476			
	similarity	0.3737	0.3520			
brkly6	rank	0.3460	0.3276	0.3381		
	similarity	0.3525	0.3305	0.3431		
eth001	rank	0.3442	0.3260	0.3519	0.3173	
	similarity	0.3522	0.3293	0.3543	0.3170	
nyuir1	rank	0.3504	0.3250	0.3282	0.3373	0.3205
	similarity	0.3559	0.3273	0.3334	0.3402	0.3230

Table 11: Average precision of the six runs selected from the TREC3 ad-hoc track with respect to each query

	westp1	*pircs1*	*vtc5s2*	*brkly6*	*eth001*	*nyuir1*
Q151	0.5309	0.5305	0.4085	0.5737	0.5333	0.5733
Q152	0.2316	0.2076	0.2638	0.2964	0.3013	0.0301
Q153	0.1610	0.1369	0.1356	0.2459	0.1656	0.0265
Q154	0.4732	0.3430	0.6559	0.4291	0.4143	0.5555
Q155	0.0838	0.0348	0.1871	0.1095	0.0945	0.1699
Q156	0.2658	0.6089	0.6408	0.1294	0.2695	0.4322
Q157	0.3950	0.1998	0.0411	0.1691	0.2417	0.2388
Q158	0.3430	0.2037	0.2340	0.3008	0.2769	0.2405
Q159	0.1727	0.1136	0.0673	0.1807	0.2426	0.2307
Q160	0.1776	0.1587	0.2805	0.0685	0.1002	0.2821
Q161	0.4906	0.4838	0.4053	0.2261	0.3772	0.5516
Q162	0.2924	0.3725	0.6900	0.2690	0.4785	0.2986
Q163	0.8566	0.8367	0.8094	0.7946	0.7265	0.7815
Q164	0.3733	0.4021	0.3688	0.3845	0.3290	0.4478
Q165	0.3462	0.3026	0.3673	0.1526	0.2035	0.2302
Q166	0.3786	0.4983	0.3842	0.4896	0.5283	0.4063
Q167	0.1709	0.0811	0.2799	0.1543	0.1283	0.0877
Q168	0.3418	0.4164	0.3413	0.2325	0.2352	0.3285
Q169	0.1421	0.1687	0.2204	0.1835	0.1792	0.2003
Q170	0.7590	0.7712	0.6956	0.6086	0.6473	0.7026
Q171	0.0430	0.0502	0.0614	0.1155	0.0883	0.0690
Q172	0.1450	0.0381	0.0972	0.0718	0.0687	0.0110
Q173	0.7678	0.6846	0.3881	0.5271	0.6404	0.7247
Q174	0.4924	0.4866	0.5524	0.4527	0.4908	0.4599
Q175	0.3496	0.3374	0.3427	0.2776	0.2845	0.3420
Q176	0.2954	0.0700	0.0554	0.0665	0.0584	0.4172
Q177	0.3461	0.2422	0.1927	0.1936	0.2032	0.1623
Q178	0.5455	0.3597	0.5057	0.4097	0.4058	0.2662
Q179	0.1682	0.0589	0.1071	0.1220	0.0652	0.0536
Q180	0.1537	0.1451	0.1068	0.2046	0.1641	0.2180
Q181	0.1082	0.0765	0.0041	0.0352	0.0308	0.0214
Q182	0.2693	0.2070	0.0213	0.2163	0.1356	0.1100
Q183	0.5438	0.1919	0.5562	0.2566	0.1594	0.2486
Q184	0.1725	0.2300	0.0483	0.1716	0.1844	0.1992
Q185	0.3246	0.7021	0.5481	0.5060	0.5769	0.5716
Q186	0.2072	0.2100	0.0826	0.1626	0.0718	0.1008
Q187	0.0695	0.0343	0.4110	0.3513	0.2330	0.0902
Q188	0.3563	0.5129	0.0669	0.1339	0.1041	0.1936
Q189	0.1542	0.3021	0.1831	0.2534	0.2862	0.3004
Q180	0.1188	0.0011	0.0405	0.0576	0.0412	0.0002
Q191	0.2621	0.1906	0.2417	0.3529	0.3207	0.1619
Q192	0.4835	0.4279	0.5167	0.5001	0.3692	0.4006
Q193	0.5082	0.6969	0.5250	0.6347	0.5302	0.4163
Q194	0.0054	0.0760	0.1130	0.0756	0.0662	0.0040
Q195	0.1341	0.0221	0.1773	0.1097	0.1310	0.0094
Q196	0.5350	0.5683	0.3745	0.4808	0.4586	0.2437
Q197	0.2588	0.2839	0.1627	0.2965	0.3248	0.2122
Q198	0.4218	0.4580	0.2363	0.4376	0.4501	0.3977
Q199	0.1886	0.3503	0.1723	0.1886	0.1639	0.0458
Q200	0.3722	0.1204	0.2026	0.2139	0.1033	0.3449
average	0.3157	0.3001	0.2914	0.2775	0.2737	0.2722

Table 12: Comparison of rank combination and similarity combination (four runs are selected from the TREC3 ad-hoc track; averages over 50 queries)

	westp1 *eth002*	*westp1* *brkly6*	*westp1* *siems1*	*eth002* *brkly6*	*eth002* *siems1*	*brkly6* *siems1*
rank	0.3495	0.3460	0.3255	0.3210	0.2847	0.2929
similarity	0.3460	0.3525	0.3166	0.3164	0.2859	0.2842

Figure 1 shows that the six runs generate slightly different rank-similarity curves. However, some runs in the TREC3 ad-hoc track generate quite different rank-similarity curves. For example, Figure 2 shows that the rank-similarity curves for four runs, namely *westp1*, *eth002*, *brkly6* and *siems1* in which the rank-similarity curves of *westp1* and *brkly6* are much different from those of *eth002* and *siems1*. We have applied the CombMNZ function to pairwise combinations of the four runs. Performance results of the combination runs are presented in Table 12. The table shows that using rank provides better retrieval effectiveness when combining two runs that generate very much different rank-similarity curves such as *westp1* & *eth002*, *westp1* & *siems1*, and *brkly6* & *siems1*.

5 Concluding Remarks

Various strategies for representing queries and documents, and various retrieval techniques are available in the IR literature. Several researchers have investigated the effect of combining multiple representations of either queries or documents, or multiple retrieval techniques on retrieval performance because different representations or different retrieval techniques can retrieve different documents. Recent work shows that significant improvements can be achieved by combining multiple evidence. However, little effort has been made to understand why multiple evidence combination results in improving retrieval effectiveness.

In this paper we analyzed why improvements can be achieved by combining evidence, and investigated how multiple evidence should be combined. We analyzed research results obtained in the data fusion literature, and gave a new rationale for evidence combination that different runs retrieve similar sets of relevant documents but retrieve different sets of nonrelevant documents. We evaluated a variety of combining methods in which the function called CombMNZ provided better retrieval effectiveness than the others. We explained the coincidence between our rationale and the properties of the CombMNZ function in that CombMNZ increases the odds of a document being at high rank based on the number of runs retrieving the document. We also investigated the effect of using rank instead of similarity on retrieval effectiveness, and found that using rank gives better retrieval effectiveness than using similarity if the runs in the combination generate quite different rank-similarity curves.

Acknowledgments

This work was supported in part by a UNDP fellowship held by the author during April to June in 1996. Additional support was provided by the NSF Center for Intelligent Information Retrieval at the University of Massachusetts at Amherst. I would like to thank Bruce Croft and Jamie Callan for discussing these works, and James Allan, Leah Larkey and Warren Greiff for reading earlier drafts of this paper. I would also like to give special thanks to NIST for making TREC results available.

References

[1] B.T. Bartell, G.W. Cottrell and R.K. Belew, "Automatic combination of multiple ranked retrieval systems," Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 173-181, 1994.

[2] N.J. Belkin, C. Cool, W.B. Croft and J.P. Callan, "The effect of multiple query representations on information retrieval performance," Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 339-346, 1993.

[3] N.J. Belkin, P. Kantor, E.A. Fox and J.A. Shaw, "Combining evidence of multiple query representation for information retrieval," Information Processing & Management, Vol. 31, No. 3, pp. 431-448, 1995.

[4] E.A. Fox and J.A. Shaw, "Combination of multiple searches," Proceedings of the 2nd Text REtrieval Conference (TREC-2), National Institute of Standards and Technology Special Publication 500-215, pp. 243-252, 1994.

[5] D. Harman, "Overview of the 1st text retrieval conference," Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.36-48, 1993.

[6] J. Katzer, M.J. McGill, J.A. Tessier, W. Frakes and P. Dasgupta, "A study of the overlap among document representations," Information Technology: Research and Development, Vol. 1, No. 2, pp. 261-274, 1982.

[7] J.H. Lee, "Combining Multiple Evidence from Different Properties of Weighting Schemes," Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 180-188, 1995.

[8] J.H. Lee, "Analyzing the Effectiveness of Extended Boolean Models in Information Retrieval," Technical Report TR95-1501, Department of Computer Science, Cornell University, Ithaca, NY, 1995.

[9] M. McGill, M. Koll and T. Norreault, "An evaluation of factors affecting document ranking by information retrieval systems," School of Information Studies, Syracuse University, 1979.

[10] T. Saracevic and P. Kantor, "A study of information seeking and retrieving. III. Searchers, searches, overlap," Journal of the American Society for Information Science, Vol. 39, No. 3, pp. 197-216, 1988.

[11] H. Turtle and W.B. Croft, "Evaluation of an inference network-based retrieval model," ACM Transactions on Information Systems, Vol. 9, No. 3, pp. 187-222, 1991.

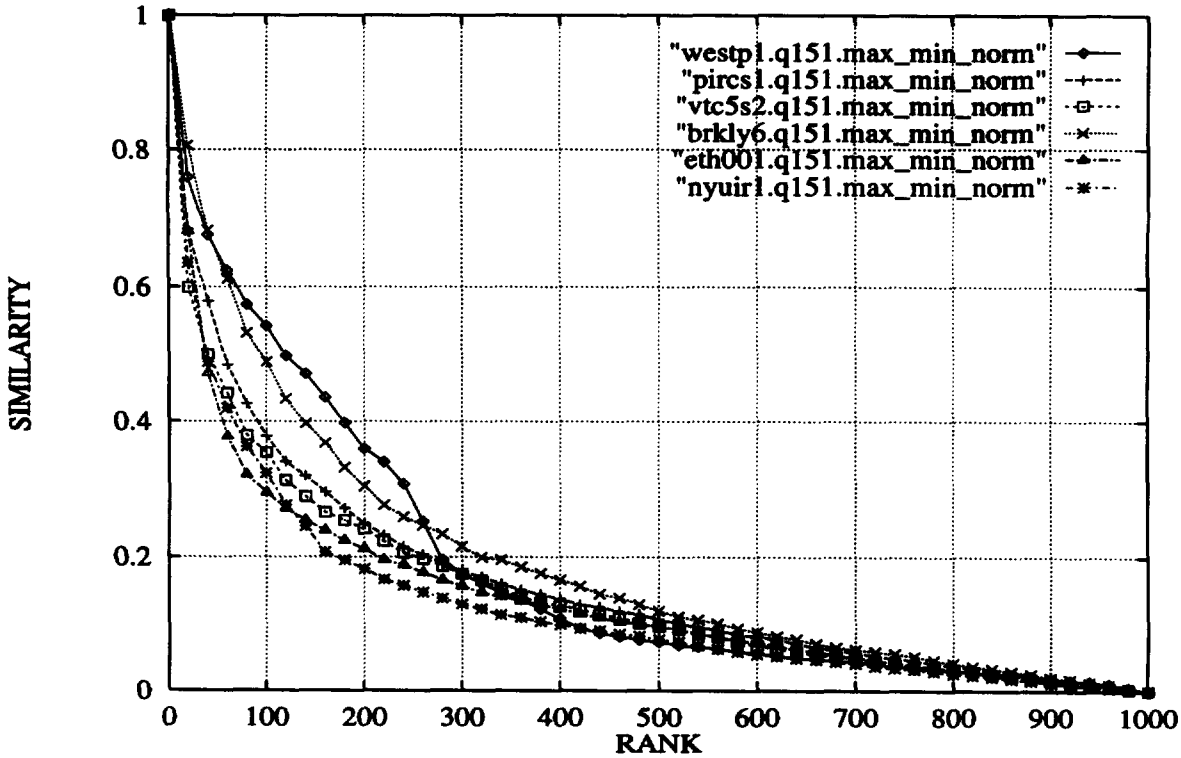

Figure 1: rank-similarity curves

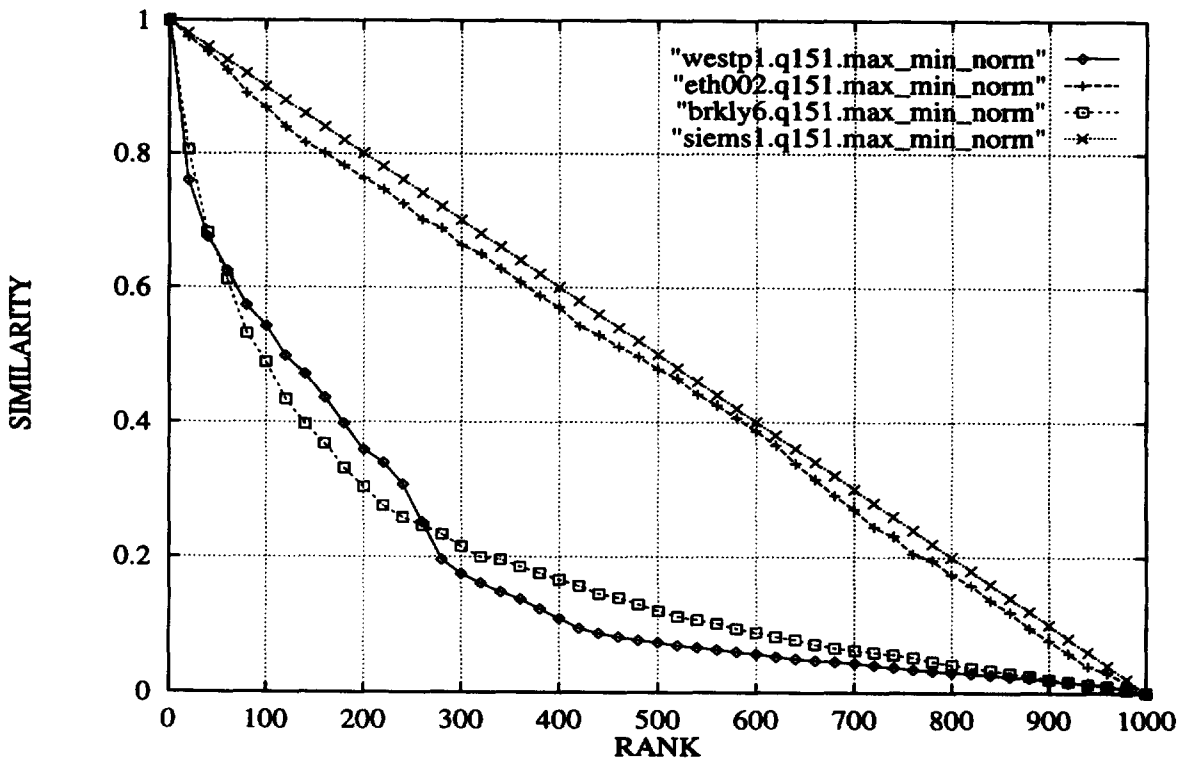

Figure 2: rank-similarity curves