*To appear in Computational Linguistics and Chinese Language Processing*

# Improving the Effectiveness of Information Retrieval with Clustering and Fusion

Jian Zhang[1]
Department of Computer Science of Tsinghua University, China
ajian@s1000e.cs.tsinghua.edu.cn

Jianfeng Gao
Microsoft Research China
jfgao@microsoft.com

Ming Zhou
Microsoft Research China
mingzhou@microsoft.com

Jiaxing Wang
Department of Computer Science of Tsinghua University, China
wjx@s1000e.cs.tsinghua.edu.cn

## ABSTRACT
Fusion and clustering are two approaches to improving the effectiveness of information retrieval. In fusion, ranked lists are combined together by various means. The motivation is that different IR systems usually emphasize different query features when determining relevance and therefore retrieve different sets of documents. In clustering, documents are clustered either before or after retrieval. The motivation is that closely associated documents tend to be relevant to the same query so that it is likely to retrieve more relevant documents by clustering. In this paper, we present a novel fusion technique that can be combined with clustering to achieve consistent improvements. Our method involves three steps: (1) clustering, (2) re-ranking, and (3) fusion. Experiments show that our approach is more efficient than conventional approaches.

## Keywords
Information retrieval, fusion, document clustering.

## 1. Introduction
In the last decade, although the overall performance of typical IR systems is not drastically increased [Voorhees 1997], for each individual query, different systems usually retrieve different sets of documents. This observation leads to the idea of combining results retrieved by different systems to improve the overall performance.

Fusion is a technique that combines results retrieved by different systems to form a single list of documents. However, conventional fusion techniques only consider retrieval results (or ranked lists), while the information embeded in the document collection is ignored. On the other hand, document clustering can acquire the structure of a document collection automatically, but it usually works on individual ranked list and does not take advantage of multiple ranked lists.

We present a novel fusion technique that can be combined with clustering. We perform clustering on each ranked list and obtain a list of clusters, and then identify clusters that contain more relevant documents. Each of those clusters is evaluated by a metric called *reliability,* and documents in the *reliable* clusters are re-ranked. Finally, a conventional fusion method is applied to produce combined results. Our experiments on TREC-5 Chinese corpus show that using this technique, we achieve consistent improvements over conventional approaches.

The remainder of this paper is organized as follows. Section 2 gives a review of related previous work. In Section 3, we describe our method in detail. In Section 4, a series of experiments are presented to show the effectiveness of our approach. The experimental results is also analysed. Finally, we present our conclusions in Section 5.

## 2. Related Work
Fusion and clustering have been two important research topics for many researchers.

Fox and Shaw ([Fox 94]) reported their work on result sets fusion. Their method for combining the evidence from multiple retrieval runs is based on document-query similarities in different sets. Five combining strategies were investigated, as summarized in Table 1. In their results, CombSUM and CombMNZ are better than others.

[1] This work was done while the author worked for Microsoft Research China as a visiting student.

| Name | Combined Similarity = |
|---|---|
| CombMAX | MAX(Individual Similarities) |
| CombMIN | MIN(Individual Similarities) |
| CombSUM | SUM(Individual Similarities) |
| CombANZ | $\frac{\text{SUM(Individual Similarities)}}{\text{Number of Nonzero Similarities}}$ |
| CombMNZ | SUM(Individual Similarities) * Number of Nonzero Similarities |

Table 1: Formulas proposed by Fox & Shaw

Thompson's work ([Thompson 90]) includes assigning each ranked list a variable weight based on the prior performance of the system. His idea is that a retrieval system should be considered more preferable than others if its prior performance is better. Thompson's result is slightly better than Fox's work.

Bartell ([Bartell 94]) used numerical optimization techniques to determine optimal scalars for a linear combination of results. The idea is similar to Thompson's except that Bartell obtained the optimal scalars from training data while Thompson constructed scalars based on their prior performance. Bartell achieved good results on a relatively small collection (less than 50MB).

To perform a more effective fusion, researchers turned to investigate whether two result sets are suitable for fusion by examining some critical characteristics. Lee ([Lee 97]) found that the overlap of the result sets was an important factor to consider in fusion. Overlap ratios of Relevant and Non-relevant documents are calculated as follows.

$$R_{overlap} = \frac{R_{common} \times 2}{R_A + R_B}$$

$$N_{overlap} = \frac{N_{common} \times 2}{N_A + N_B}$$

where $R_A$ and $N_A$ are, respectively, the numbers of relevant and irrelevant documents in result set $RL_A$[2]. $R_{common}$ is the number of common relevant documents in $RL_A$ and $RL_B$. $N_{common}$ is the number of common irrelevant documents in $RL_A$ and $RL_B$.

Lee observed that fusion works well for result sets that have a high $R_{overlap}$ and a low $N_{overlap}$. Inspired by this observation, we will also incorporate $R_{common}$ in our fusion approach.

Vogt tested different linear combinations of several results from TREC-5. 36,600 result pairs were tested. A linear regression of several potential indicators was performed to determine the potential improvement for result sets to be fused. Thirteen factors including measures of individual inputs such as average precision/recall and some pairwise factors such as overlap and unique document counts were considered. Vogt concluded that the characteristics for effective fusion are (1) at least one result has high precision/recall, (2) a high overlap of relevant documents and a low overlap of non-relevant documents, (3) similar distribution of relevance scores, and (4) each retrieval system ranks relevant documents differently. Conclusion (1) and (2) are also confirmed by our experiments, as will be shown in Section 4.3.

Clustering is now considered as a useful method in information retrieval for not only documents categorization but also interactive retrieval. The use of clustering in information retrieval is based on the *Clustering Hypothesis* "closely associated documents tend to be relevant to the same requests"([Rijsbergen, 1979]). Hearst ([Hearst 96]) showed that this hypothesis holds on a set of documents returned by a retrieval system. According to this hypothesis, if we do a good job at clustering the retrieved documents, we are likely to separate the relevant and non-relevant documents into different groups. If we can direct the user to the correct group of documents, we would enhance the chance of finding interesting information for the user. Previous works ([Cutting et al, 1992], [Anton 2000]) focused on clustering documents and let the user select the cluster they are interested. Their approaches are interactive. Most clustering methods abovementioned work on individual ranked list, and do not take any advantage of multiple ranked lists.

[2] $RL_A$ means ranked list returned by retrieval system A.

In this paper, we combine clustering with fusion. Our approach differs from interactive approaches in three aspects. First, we use two or more ranked lists while others usually use one in clustering. Second, user interactive input is not needed in our approach. Third, we provide a ranked list of documents to the user instead of a set of clusters.

## 3. Fusion with Clustering

Our method is based on two hypotheses:

***Clustering Hypothesis:*** Documents that are relevant to the same query can be clustered together since they tend to be more similar to each other than to non-relevant documents.

***Fusion Hypothesis:*** Different ranked lists usually have a high overlap of relevant documents and a low overlap of non-relevant documents.

*Clustering Hypothesis* suggests that we might be able to roughly separate relevant documents from non-relevant documents with a proper clustering algorithm. Relevant documents can be clustered into one or several clusters, and these clusters contain more relevant documents than others. We call such a cluster a *reliable cluster*.

*Fusion hypothesis* brings us the idea of identifying *reliable clusters*. The *reliable clusters* from different ranked lists usually have a high overlap. Therefore, the more relevant documents a cluster contains, the more reliable the cluster is. We will describe the computation of *reliability* in detail in Section 3.3.

Figure 1 shows the basis idea of our approach. Two clusters (a1 and b1) from different ranked lists that have the largest overlap are identified to be the reliable clusters.

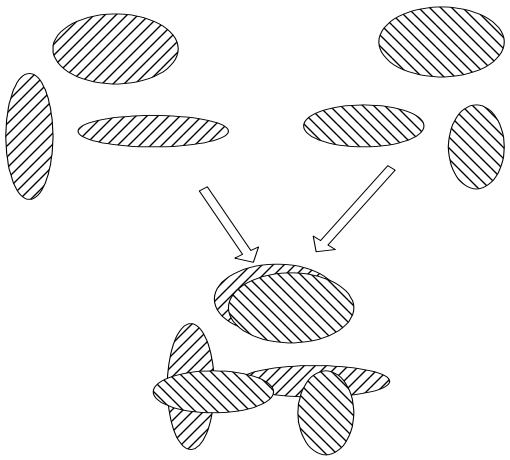

Fig1: Clustering results of two ranked lists

Our approach consists of three steps. First, we cluster each ranked list. Then we identify the *reliable clusters* and adjust the relevance value of each document according to the *reliability* of the cluster. We finally use CombSUM to combine the adjusted ranked lists and present to user.

In the following sections, we will describe our approach in more detail. For conciseness, we use some symbols to present our approach, which are listed in Table 2 with their explanation.

| Symbol | Explanation |
|---|---|
| $q$ | A query |
| $d$ | A document |
| $RL_A, RL_B$ | Ranked list returned by retrieval systems A and B, respectively |
| $C_{A,i}$ | $i$ th cluster in $RL_A$ |
| $Sim\_CC(C_{A,i}, C_{B,j})$ | Similarity between $C_{A,i}$ and $C_{B,j}$ |

| | |
|---|---|
| $Sim\_qC(q, C_{A,i})$ | Similarity between query $q$ and $C_{A,i}$ |
| $Sim\_dd(d_i, d_j)$ | Similarity between two documents, $d_i$ and $d_j$ |
| $r(C_{A,i})$ | *Reliability* of cluster $C_{A,i}$ |
| $rel_A(d)$ | Relevance score of document $d$ given by retrieval system A |
| $rel_A^*(d)$ | Adjusted relevance score of document $d$ |
| $rel(d)$ | Final relevance score of document $d$ |

Table 2: Notations

## 3.1 Clustering

The goal of clustering is to separate relevant documents from non-relevant documents. To accomplish this, we need to define a measure for the similarity between documents and design corresponding clustering algorithm.

### *3.1.1 Similarity between documents*

In our experiments, we use vector space model to represent documents. Each document is represented as a vector of weights $(w_{i1}, w_{i2}, ...w_{im})$, where $w_{ik}$ is the weight of the term $t_k$ in document $d_i$. The weight $w_{ik}$ is determined by the occurrence frequency of $t_k$ in document $d_i$ and its distribution in the entire collection. More precisely, the following formula is used to compute $w_{ik}$.

$$w_{ik} = \frac{[\log(f_{ik}) + 1.0] \times \log(N / n_k)}{\sqrt{\sum_j [(\log(f_{ij}) + 1.0) \times \log(N / n_j)]^2}} \quad (1)$$

where $f_{ik}$ is the occurrence frequency of the term $t_k$ in the document $d_i$, N is the total number of documents in the collection and $n_k$ is the number of documents that contain the term $t_k$. Actually, this is one of the most frequently used *tf*idf* weighting schemes in IR.

For any two documents $d_i$ and $d_j$, the cosine measure as given below is used to determine their similarity.

$$Sim\_dd(d_i, d_j) = \frac{\sum_k (w_{ik} \times w_{jk})}{\sqrt{\sum_k w_{ik}^2 \times \sum_k w_{jk}^2}} \quad (2)$$

### *3.1.2 Clustering algorithm*

There are many clustering algorithms for document clustering. Our task is to cluster a small collection of documents returned by individual retrieval systems. Since the size of the collection is 1,000 in our experiments, the complexity of the clustering algorithm is not a serious problem.

Figure 2 shows our clustering algorithm. The LoopThreshold and ShiftThreshold value are set to 10 in our experiments.

---

Randomly set document $d_i$ to cluster $C_j$;

LoopCount =0; ShiftCount = 1000;

While (LoopCount < LoopThreshold and ShiftCount > ShiftThreshold) Do

Construct the centroid of each cluster, i.e.

Centroid of $C_j = \frac{\sum_{d_i \in C_j} d_i}{|C_j|}$;

Assign $d_i$ to its nearest cluster(the distance is determined by the similarity between $d_i$ and the centroid of cluster);

---

ShiftCount = the number of documents shift to other cluster;

LoopCount++;

Figure 2: algorithm documents clustering

The ideal result is that clustering can gather all relevant documents into one cluster, and all non-relevant documents into the others. However, this is unlikely to happen. In fact, relevant documents are usually distributed in several clusters. After clustering, each ranked list is composed of a set of clusters, say $C_1, C_2 \ldots C_n$.

### *3.1.3 Size of cluster*

The size of cluster is the number of documents in a cluster. The clustering algorithm in Figure 2 cannot guarantee that the clusters are of identical size. This causes many problems because the overlap depends on the size of each cluster.

To solve this problem[3], we restrict the clusters to the same size with following approach. For clusters that contain more documents than the average, we remove the documents that are far from the cluster's centroid. These removed documents are added to clusters that are smaller than the average.

Since all clusters are of the same size, the size of cluster becomes a parameter in our algorithm. Thus we need to set this parameter to an optimal value for the best performance. We will report the experiments for this purpose in Section 4.3.

## 3.2 Re-ranking

After clustering each ranked list, we obtain a group of clusters each of which contains more or less relevant documents. By re-ranking, we expect to determine *reliable clusters* and adjust the relevance score of documents in each ranked list such that the relevance scores become more reasonable. To identify *reliable clusters*, we assign each cluster a *reliability* score. According to the *Fusion Hypothesis*, we use the overlap between clusters to compute the *reliability* of a cluster. The *reliability* $r(C_{A,i})$ of cluster $C_{A,i}$ is computed as follows (see Table 2 for the meaning of the symbols):

$$r(C_{A,i}) = \sum_j \left[ \frac{Sim\_qC(q, C_{B,j})}{\sum_t Sim\_qC(q, C_{B,t})} Sim\_CC(C_{A,i}, C_{B,j}) \right] \tag{3}$$

Where

$$Sim\_CC(C_{A,i}, C_{B,j}) = \left| C_{A,i} \cap C_{B,j} \right| \tag{4}$$

$$Sim\_qC(q, C_{A,i}) = \frac{\sum_{d \in C_{A,i}} rel_A(d)}{\left| C_{A,i} \right|} \tag{5}$$

In equation (4), the similarity of two clusters is estimated in terms of the common documents they both contain. In equation (5), similarity between query and cluster is estimated in terms of the average relevance score of the documents that the cluster contains. In equation (3), for each cluster $C_{A,i}$ in $RL_A$, its reliability $r(C_{A,i})$ is defined to be the weighted sum of the similarity between the cluster $C_{Ai}$ and all clusters in $RL_B$. The intuition underlying this formula is that the more similar two clusters are, the more reliable they are, as illustrated in Figure 1.

Since *reliability* represents the precision of a cluster, we use it to adjust the relevance score of documents in each cluster. Formula (6) adjusts the relevance score of a document in a high reliable cluster.

[3] The size of cluster and the number of clusters are critical issues, which have been studied by many researchers. This paper is focused on how to combine fusion and clustering together and show some potential capability of the combination. So we use very simple method to solve the problem. Our clustering algorithm is also very simple. It remains to be our future work to investigate the impact of different algorithms.

$$rel_A^*(d) = rel_A(d) \times [1 + r(C_{A,t})] \quad (6)$$

where $d \in C_{A,t}$

## 3.3 Fusion

Each original ranked list has been adjusted by clustering and re-ranking. We now combine these improved ranked lists together using the following formula (i.e. CombSUM in [Fox 94]):

$$rel(d) = rel_A^*(d) + rel_B^*(d) \quad (7)$$

In equation (7), the combined relevance of document $d$ is the sum of each adjusted relevance values that have been computed in the previous steps.

# 4. Experimental Results

In this section, we present the results of our experiments. We first describe our experimental settings in Section 4.1. In Section 4.2, we verify the two hypotheses described in Section 3 with some experiments. In Section 4.3, we compare our approach with other three conventional fusion methods. Finally, we examine the impact of cluster size.

## 4.1 Experiment settings

We use several retrieval results from TREC-5 Chinese information retrieval track in our fusion experiments. The documents collection contains articles published in the People's Daily and news released by the Xinhua News Agency. Some statistical characteristics of the collection are summarized in Tables 3.

| | |
|---|---|
| Number of docs | 164,811 |
| Total size (Mega Bytes) | 170 |
| Average doc length (Characters) | 507 |
| Number of queries | 28 |
| Average query length (Characters) | 119 |
| Average Number of rel docs / query | 93 |

Table 3 Characteristics of TREC-5 Chinese Corpus

The 10 groups who took part in TREC-5 Chinese provided 20 retrieval results. We randomly pick seven ranked lists for our fusion experiments. The tags and average precision are listed in Table 4. It is noticed that the average precision is similar except HIN300.

| Ranked list | AvP (11 pt) |
|---|---|
| BrklyCH1 | 0.3568 |
| CLCHNA | 0.2702 |
| Cor5C1vt | 0.3647 |
| HIN300 | 0.1636 |
| City96c1 | 0.3256 |
| Gmu96ca1 | 0.3218 |
| gmu96cm1 | 0.3579 |
| *Average :* | *0.3086* |

Table 4: Average precision of individual retrieval system

Since ranges of similarity values of different retrieval results are quite different, we normalize each retrieval result before combining them. The bound of each retrieval result is mapped to [0,1] with the following formula [Lee 97].

$$normalized\_rel = \frac{unnormalized\_rel - minimum\_rel}{maximum\_rel - minimum\_rel}$$

## 4.2 Examining the hypotheses

We first examine the two hypotheses we mentioned in Section 3.

In relation to *Clustering Hypothesis*, we cluster each ranked list into 10 clusters with our clustering algorithm. Table 5 shows some statistical information of the clustering results. The first row lists four kinds of clusters containing non, 1, 2-10 and more than 10 relevant

document(s). The second row shows corresponding percentage of each kind of clusters. The third row shows the percentage of relevant documents in each kind of cluster. From this table, we can draw 2 observations. First, there are about 50% clusters that contain 1 or no relevant document. Second, most relevant documents (more than 60%) are in a small number of clusters (about 7%). According to these observations, we can draw a conclusion that relevant documents are concentrated in a few clusters.

Thus the *Clustering Hypothesis* holds in terms of the initial retrieval result when a proper algorithm is adopted.

| Different kinds of clusters | Containing no relevant doc | Containing 1 relevant doc | Containing 2-10 relevant docs | Containing >10 relevant docs |
|---|---|---|---|---|
| Percentage of this kind of clusters | 38.3% | 15.0% | 35.0% | 7.0% |
| Percentage of relevant docs contained in this kind of clusters | 0% | 3.7% | 35.8% | 60.5% |

Table 5: distribution of relevant docs

To test the *Fusion Hypothesis*, we compute $R_{overlap}$ and $N_{overlap}$ for each combination pairs. Table 6 lists some results. The last row shows that the average $R_{overlap}$ is 0.7688 while the corresponding average $N_{overlap}$ is 0.3351. It turns out that *Fusion Hypothesis* holds on the retrieval results we used.

Table 6 will also be used in Section 4.3 to explain that $R_{overlap}$ is the most important factor that determines the performance of fusion. We mark those rows whose $R_{overlap}$ score is higher than 0.80 with character *.

| Combination pair | $R_{overlap}$ | $N_{overlap}$ |
|---|---|---|
| BrklyCH1 & CLCHNA | * 0.8542 | 0.3398 |
| BrklyCH1 & Cor5C1vt | * 0.9090 | 0.4393 |
| BrklyCH1 & HIN300 | 0.4985 | 0.2575 |
| BrklyCH1 & City96c1 | * 0.8996 | 0.4049 |
| BrklyCH1 & Gmu96ca1 | * 0.8784 | 0.3259 |
| BrklyCH1 & gmu96cm1 | * 0.8871 | 0.3292 |
| CLCHNA & Cor5C1vt | * 0.8728 | 0.4118 |
| CLCHNA & HIN300 | 0.4652 | 0.2172 |
| CLCHNA & City96c1 | * 0.8261 | 0.2668 |
| CLCHNA & Gmu96ca1 | * 0.8447 | 0.3090 |
| CLCHNA & gmu96cm1 | * 0.8585 | 0.3412 |
| Cor5C1vt & HIN300 | 0.4961 | 0.2392 |
| Cor5C1vt & City96c1 | * 0.8763 | 0.2943 |
| Cor5C1vt & Gmu96ca1 | * 0.9193 | 0.4742 |
| Cor5C1vt & gmu96cm1 | * 0.9185 | 0.4525 |
| HIN300 & City96c1 | 0.4813 | 0.1555 |
| HIN300 & Gmu96ca1 | 0.4636 | 0.1854 |
| HIN300 & gmu96cm1 | 0.4701 | 0.2004 |
| City96c1 & Gmu96ca1 | * 0.8698 | 0.2854 |
| City96c1 & gmu96cm1 | * 0.8860 | 0.3005 |
| Gmu96ca1 & gmu96cm1 | * 0.9687 | 0.8064 |
| *Average* | *0.7688* | *0.3351* |

Table 6: $R_{overlap}$ and $N_{overlap}$ of combination pairs

## 4.3 Comparison with conventional fusion methods

First, we study three combination methods that were proposed by [Fox 94], namely, CombMAX, CombSUM, and CombMNZ. Their fusion results on the same data set are listed in Table 7. The last row is the average precision of all combination pairs. Since the average precision of the seven retrieve results is 0.3086 (see Table 4), each of these three fusion methods has improved the average precision significantly. It looks that CombSUM is the best one among them. This confirms the observation in [Fox 94].

Then we compare our approach with these three methods, as shown in the last column in Table 7. Our new approach brings 3% improvement beyond CombSUM. We also find that among all the 21 combination pairs, 17 of them are improved, compared to the CombSUM approach. We mark these rows with a character *.

| Combination pair | CombMAX | CombSUM | CombMNZ | Our Approach (Cluster size=100) |
|---|---|---|---|---|
| BrklyCH1 & CLCHNA | 0.3401 | 0.3627 | 0.3549 | * 0.3755 |
| BrklyCH1 & Cor5C1vt | 0.3832 | 0.3976 | 0.3961 | * 0.4107 |
| BrklyCH1 & HIN300 | 0.3560 | 0.3243 | 0.2618 | 0.3107 |
| BrklyCH1 & city96c1 | 0.3650 | 0.3833 | 0.3856 | * 0.3912 |
| BrklyCH1 & gmu96ca1 | 0.3753 | 0.4028 | 0.3999 | * 0.4022 |
| BrklyCH1 & gmu96cm1 | 0.3979 | 0.4234 | 0.4201 | * 0.4243 |
| CLCHNA & Cor5C1vt | 0.3434 | 0.3560 | 0.3492 | * 0.3707 |
| CLCHNA & HIN300 | 0.2746 | 0.2478 | 0.2154 | 0.2579 |
| CLCHNA & city96c1 | 0.3007 | 0.3459 | 0.3573 | * 0.3931 |
| CLCHNA & gmu96ca1 | 0.3269 | 0.3667 | 0.3634 | * 0.3690 |
| CLCHNA & gmu96cm1 | 0.3555 | 0.3864 | 0.3783 | * 0.3883 |
| Cor5C1vt & HIN300 | 0.3778 | 0.3081 | 0.2520 | 0.3139 |
| Cor5C1vt & city96c1 | 0.3709 | 0.4091 | 0.4104 | * 0.4285 |
| Cor5C1vt & gmu96ca1 | 0.3568 | 0.3684 | 0.3676 | * 0.3724 |
| Cor5C1vt & gmu96cm1 | 0.3831 | 0.3926 | 0.3911 | * 0.3975 |
| HIN300 & city96c1 | 0.2616 | 0.2565 | 0.2444 | 0.3036 |
| HIN300 & gmu96ca1 | 0.3466 | 0.2942 | 0.2464 | 0.2954 |
| HIN300 & gmu96cm1 | 0.3764 | 0.3205 | 0.2613 | 0.3150 |
| city96c1 & gmu96ca1 | 0.3310 | 0.3764 | 0.3854 | * 0.3939 |
| city96c1 & gmu96cm1 | 0.3595 | 0.3970 | 0.4047 | * 0.4090 |
| gmu96ca1 & gmu96cm1 | 0.3451 | 0.3514 | 0.3511 | * 0.3505 |
| *Average:* | *0.3489* | *0.3557* | *0.3426* | *0.3654* |

Table 7: average precision of each combination pairs

Comparing the results in Table 7 with that in Table 6, we find that the pairs with a $R_{overlap}$ over 0.80 lead to better combination performance. We call this kind of pair *combinable pair*. For example, BrklyCH1 & CLCHNA is a *combinable pair*. Although the average combination performance is 0.3654 (using our approach), almost all *combinable pairs* exceed the average performance[4]. This, again confirms the conclusion in both [Lee 97] and [Vogt 98] that the performance of fusion heavily depends on $R_{overlap}$. It also reveals the limitation of our approach and other linear fusion techniques that a high overlap of relevant document is a pre-requisite for performance enhancement. For those pairs that don't fulfill this pre-requisite, normal fusion can even decrease the retrieval performance.

Since ranked lists are combined linearly, only the ratio of the two weights affects the final performance.

$$RL_{combined} = RL_A + wRL_B$$

[4] "gmu96ca1 & gmu96cm1" is an exception because their related $N_{overlap}$ score is very high.

CombSUM can be taken as a special case of linear combination where w is set to be 1. When the relevant documents are known, the weight w can be optimized using some numerical method. In our experiment, the weight w is optimized using golden section search ([William 95]). This approach is adopted in [Vogt 98]. The average precision for the best linear combination we got is 0.3714. As shown in Fig 3, our approach is better than CombSUM and CombMAX and reaches very closely to CombBest.

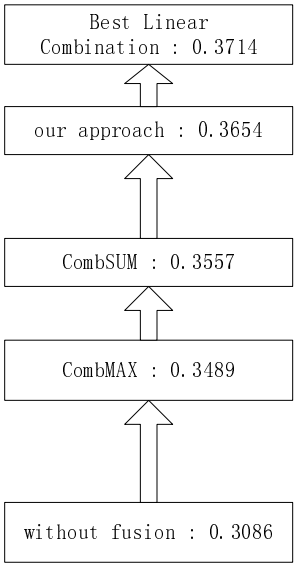


Figure 3 Performance of different approaches

To summarize, we had three conclusions from the above experiments. First, in most cases, our new approach shows a better performance than most conventional methods, including CombSUM and CombMNZ. Second, $R_{overlap}$ strongly affects the performance of linear fusion. Third, our approach has a performance very close to the best linear combination.

## 4.4 Impact of cluster size

We also study the impact of cluster size. Table 8 shows the experimental results. When cluster size varies from 200 to 5, the average precision doesn't change much. The maximum value is 0.3675 when cluster size is 25 and the minimum value is 0.3621 when cluster size is 200. It shows that cluster size setting has very little impact on our approach.

| Size of Cluster | 200 | 100 | 50 | 25 | 10 | 5 |
|---|---|---|---|---|---|---|
| 11pt AvP | 0.3621 | 0.3654 | 0.3661 | 0.3675 | 0.3668 | 0.3661 |

Table 8: Impact of cluster size

Another interesting question is what will happen when cluster size is set to 1000 or 1.

When cluster size is set to 1000, each ranked list becomes a single cluster. Then the reliability of $C_A$ and $C_B$ can be computed as follows:

$$r(C_A) = r(C_B) = Sim\_CC(C_A, C_B) = \left|C_A \cap C_B\right|$$

Since $r(C_A)$ and $r(C_B)$ are equal, the re-ranking and fusion progress turns to be a normal CombSUM step and the average precision is equal to that of the CombSUM approach.

When cluster size is set to 1, each document alone forms a cluster itself. Those documents appearing in both ranked lists will be improved. For those documents that only appear in one ranked list, their relevance remains unchanged. On the other hand, the relevance score of those documents that appear in both ranked lists will be improved with a factor of $1 + \frac{Sim\_dd(q,d)}{\sum Sim\_dd(q,d_j)}$. The final result will be close to that of CombSUM approach because this factor is close to 1.

The impact of cluster size setting is illustrated in Figure 4. According to this figure, we find that fusion combined with clustering is consistently better than the approaches without clustering (where cluster size = 1000). We find that setting size to 25 gives the best combination when ranked list has a size of 1,000.

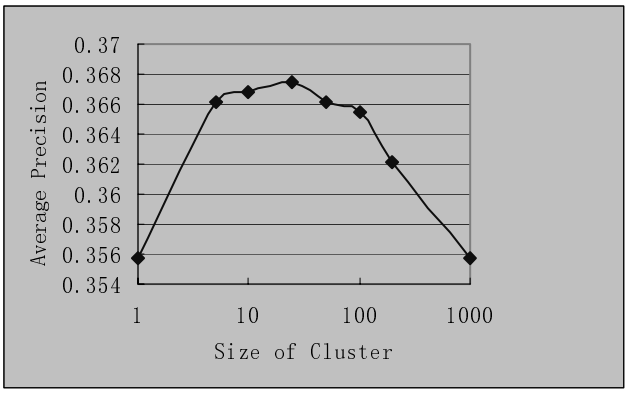


Figure 4 Impact of size of cluster

## 5. Conclusion

Combining multiple retrieval results is certainly a practical technique to improve the overall performance of an information retrieval system. In this paper, we propose a novel fusion method that can be combined with document clustering to improve the retrieval effectiveness. Our approach consists of three steps. First, we apply clustering to the initial ranked document lists to obtain a list of document clusters. Then we identify reliable clusters and adjust each ranked list separately by our re-ranking approach. Finally a conventional fusion is done to produce an adjusted ranked list.

Since our approach is based on two hypotheses, we first verified them by experiments. We also compared our approach with other conventional approaches. The results showed that each of them brings some improvements and our approach compared favorably with them. We also investigated the impact of cluster size. We found that our approach is rather stable over the change size of cluster.

Although our method shows good performance in experiments, we believe it still can be further improved. A better clustering algorithm to identify more reliable clusters and more elaborated formula to re-rank ranked list are expected to bring up the improvement. These are the topics for our future work.

## 6. References

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

[Cutting 1992] D.R.Cutting, D.R.Karger, J.O.Pedersen, and J.W.Tukey. "Scatter/gather: A cluster-based approach to browsing large document collections". In *Proceedings 15th Annual International ACM-SIGIR Conference*, 1992, pages 126-135.

[Diamond 1996] T.Diamond. "Information retrieval using dynamic evidence combination". *PhD Dissertation Proposal*. 1996.

[Voorhees 1997] E.Voorhees, D.Harman. "Overview of the Sixth Text Retrieval Conference (TREC-6)", *NIST Special Publication 500-240*, 1997.

[Fox 1994] Fox,E. and J.Shaw. "Combination of Multiple searches", *Proceedings of the 2nd Text Retrieval Conference (TREC2)*, NIST Special Publication 500-215, 1994.

[Hearst 1996] Hearst,M. and Pedersen,P. "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results", *Proceedings of 19th Annual International ACM/SIGIR Conference*, Zurich, 1996.

[Kantor 1995] Kantor,P.B." Decision level data fusion for routing of documents in the TREC3 context: A best case analysis of worst case results". *NIST Special Publication 500-226*, 1995.

[Knaus 1995] Knaus,D., Mittendorf,E., and Schauble,P. "Improving a basic retrieval method by links and passage level evidence". *NIST Special Publication 500-226*, 1995

[Lee 1997] J.H. Lee. "Analyses of multiple evidence combination.", *Proceedings of the 20th Annual International ACM-SIGIR Conference* , 1997, pages 267-276.

[Leuski 1999] A.Leuski and J.Allan. "The Best of Both Worlds: Combining Ranked List and Clustering". *CIIR Technical Report IR-172*. 1999.

[Leuski 2000] A.Leuski and J.Allan. "Improving Interactive Retrieval by Combining Ranked List and Clustering," In the *Proceedings of RIAO 2000 Conference*, Paris, 2000, pp. 665-681.

[Rijsbergen 1979] C.J.van Rijsbergen. "*Information Retrieva"l.* Butterworths, London, second edition, 1979.

[Salton 1971] G.Salton. "Cluster search strategies and the optimization of retrieval effectiveness". In G. Salton, editor, *The SMART Retrieval System*, pages 223-242. Prentice Hall Englewood Cliffs, N.J., 1971.

[Selberg 1996] Selberg, E. and Etzioni, O. “Multi-service search and comparison using the MetaCrawler”. In *Proceedings of the 4th International World Wide Web Conference*, 1996, pages 195-208.

[Shaw and Fox 1995] Shaw, J. and Fox, E. “Combination of multiple searches”. *NIST Special Publication* 500-226, 1995.

[Thompson 1990] Thompson,P. “A combination of Expert Opinion Approach to Probabilistic Information Retrieval, part I: The Conceptual Model”. *Information Processing and Management*, vol 26(3) 1990.

[Vogt 1997] Vogt, C., Cottrell, G., Belew, R., and Bartell, B. “Using relevance to train a linear mixture of experts”. In *the Fifth Text Retrieval Conference*, Ed. D. Harman, Gaitherberg, MD, 1996.

[Vogt 1998] Vogt, C. and G. Cottrell., “Predicting the Performance of Linearly Combined IR Systems,” *Proceedings of the 21st Annual International ACM SIGIR Conference,* 1998.

[Vogt 1999] Christopher C. Vogt and Garrison W. Cottrell. “Fusion Via a Linear Combination of Scores”. *Information Retrieval*, 1999.

[Willett 1988] P.Willett. “Recent trends in hierarchical document clustering: A critical review”. *Information Processing & Management*, 24(5): 577-597, 1988.

[William 1995] William H. Press, Saul A. Teukolsky, William T. Vettering, and Brian P. Flannery. “Numerical Recipes in C: The Art of Scientific Computing” Cambridge University Press, 1995.