Computational Linguistics and Chinese Language Processing
Vol. 6, No. 1, February 2001, pp. 109-125

Improving the Effectiveness of Information Retrieval with Clustering and Fusion

Jian Zhang[*], Jianfeng Gao[+], Ming Zhou[**], Jiaxing Wang[++]

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 will complement each other, because they usually emphasize different query features when determining relevance and retrieve different sets of documents. In clustering, documents are clustered either before or after retrieval. The motivation is that similar documents tend to be relevant to the same query so that this approach is likely to retrieve more relevant documents by identifying clusters of similar documents. In this paper, we present a novel fusion technique that can be combined with clustering to achieve consistent improvements over conventional approaches. Our method involves three steps: (1) clustering similar documents, (2) re-ranking retrieval results, and (3) combining retrieval results.

1. Introduction

In terms of the overall performance on a large query set, none of the typical IR systems outperform others substantially, while for each individual query, the performance that different systems achieve varies greatly [Voorhees 1997]. This observation leads to the idea of combining results obtained by different IR systems to improve overall performance.

Fusion is a technique that combines retrieval results (or ranked lists) obtained by

[*] This work was done while the author worked for Microsoft Research Asia as a visiting student. Department of Computer Science and Technology of Tsinghua University, China E-mail:ajian@s1000e.cs.tsinghua.edu.cn
[+] Microsoft Research Asia E-mail:jfgao@microsoft.com
[**] Microsoft Research Asia E-mail:mingzhou@microsoft.com
[++] Department of Computer Science and Technology of Tsinghua University, China E-mail:wjx@s1000e.cs.tsinghua.edu.cn

different systems. However, conventional fusion techniques only consider retrieval results, while the information embedded in the document collection (e.g. the similarity between documents) is ignored. On the other hand, document clustering applies the structure of a document collection, but it usually considers each individual ranked list separately and is not able to take advantage of multiple ranked lists.

In this paper, we present a novel fusion technique that can be combined with clustering. Given multiple retrieval results obtained by different IR systems, we first perform clustering on each ranked list and obtain a set of clusters. We then identify the clusters that contain the most relevant documents. Each of these clusters is evaluated based on a metric called *reliability*. Documents in *reliable* clusters are re-ranked. That is, we set higher scores for these documents. Finally, a conventional fusion method is applied to combine multiple retrieval results, which are re-ranked. Our experiments on the TREC-5 Chinese collection show that the above approach achieves consistent improvements over conventional approaches.

The remainder of this paper is organized as follows. Section 2 gives a brief survey of related 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. Finally, we present our conclusions in Section 5.

2. Related Work

Fusion and clustering have been important research topics for many researchers.

Fox and Shaw [Fox 1994] reported on 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 experiments, CombSUM and CombMNZ were better than the others.

Table 1. *Formulas proposed by Fox & Shaw.*

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

Thompson's work [Thompson 1990] includes assigning to each ranked list a variable weight based on the prior performance of the system. His idea is that a retrieval system should be considered preferable to others if its prior performance is better. Thompson's results were slightly better than Fox's.

Bartell [Bartell 1994] used numerical optimization techniques to determine optimal scalars (weights) 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 fusion more effectively, researchers began to investigate whether two result sets are suitable for fusion by examining some critical characteristics. Lee [Lee 1997] found that the overlap of the result sets was an important factor for 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 [1]. 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.

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

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 also incorporate R_{common} into our fusion approach.

Vogt [Vogt 1998, 1999] 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 distributions 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 to be a useful information retrieval method for not only documents categorization but also interactive retrieval. The use of clustering in information retrieval is based on the Clustering Hypothesis [Rijsbergen, 1979]: *"closely associated documents tend to be relevant to the same requests"*. Hearst [Hearst 1996] showed that this hypothesis holds for a set of documents returned by a retrieval system. According to this hypothesis, if we do a good job of clustering the retrieved documents, we will likely separate the relevant and non-relevant documents into different groups. If we can direct the user to the correct group of documents, we can enhance the likelihood of finding interesting information for the user. Previous works [Cutting *et al*, 1992], [Leuski 1999] and [Leuski 2000] focused on clustering documents and let users select the clusters they were interested in. Their approaches are interactive. Most of the clustering methods mentioned above work on individual ranked lists and do not take advantage of multiple ranked lists.

In this paper, we combine clustering with fusion. Our approach differs from interactive approaches in three ways. 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.

The *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 will contain more relevant documents than others. We call such a cluster a *reliable cluster*.

The *Fusion hypothesis* presents 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.

Fig.1 shows the basis idea behind our approach. Two clusters (a1 and b1) from different ranked lists that have the largest overlap are identified as reliable clusters.

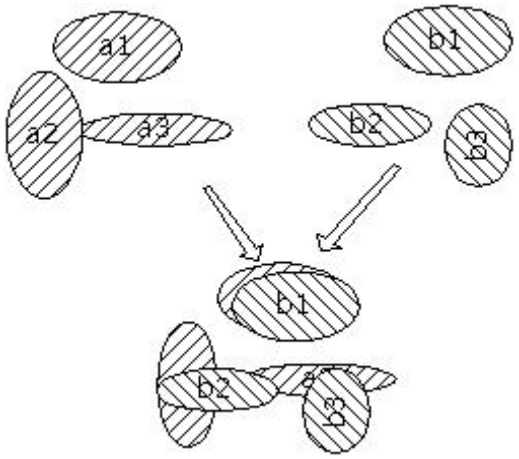

Figure 1 *Clustering results of two ranked lists.*

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. Finally, we use CombSUM to combine the adjusted ranked lists and present the result to user.

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

Table 2. *Notations.*

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

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 a corresponding clustering algorithm.

3.1.1 Similarity between documents

In our experiments, we used the 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 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 term t_k in document d_i, N is the total number of documents in the collection and n_k is the number of documents that contain 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 goal is to cluster a small collection of documents returned by an individual retrieval system. Since the size of the collection was 1,000 in our experiments, the complexity of the clustering algorithm was not a serious problem.

Fig.2 shows our clustering algorithm. The LoopThreshold and ShiftThreshold value were 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 the cluster);

ShiftCount = the number of documents shifted to other cluster;

LoopCount++;

Figure 2 *Algorithm for document clustering.*

The ideal result is obtained when clustering gathers all relevant documents into one cluster and all non-relevant documents into the other cluster. 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 a cluster

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

To solve this problem, we force the clusters to have the same size using the following approach. For clusters that contain a larger number of 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 average[2].

Since all the clusters are of the same size, the size of a cluster becomes a parameter in our algorithm. Thus, we need to set this parameter to an optimal value to achieve the best performance. We will report experiments conducted to determine this value 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. Through re-ranking, we expect to determine *reliable clusters* and adjust the relevance scores of the documents in each ranked list such that the relevance scores become more reasonable. To identify *reliable clusters*, we assign to 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 definitions 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

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

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

[2] The size of a cluster and the number of clusters are critical issues in clustering and have been studied by many researchers. This paper focuses on how to combine fusion and clustering together and shows the potential of this combination approach. Therefore, we use a very simple method to solve the problem. Our clustering algorithm is also very simple. Our future work will be to investigate the impacts of different algorithms.

In equation (4), the similarity of two clusters is estimated based on the common documents they both contain. In equation (5), the similarity between a query and a cluster is estimated based on 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 as the weighted sum of the similarity between cluster C_{Ai} and all the 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 Fig.1.

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

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

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

3.3 Fusion

So far, each original ranked list has been adjusted by means of clustering and re-ranking. We next combine these improved ranked lists together using the following formula (i.e. CombSUM in [Fox 1994]):

$$rel(d) = rel_A^*(d) + rel_B^*(d). \tag{7}$$

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

4. Experimental Results

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

4.1 Experiment settings

We used several retrieval results from the TREC-5 Chinese information retrieval track in our fusion experiments. The document 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.

Table 3. *Characteristics of the TREC-5 Chinese collection.*

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 relevant docs/query	93

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

Table 4. *Average precision of individual retrieval system*

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*

Since the ranges of similarity values of the different retrieval results were quite different, we normalized each retrieval result before combining them. The bound of each retrieval result was mapped to [0,1] using the following formula [Lee 1997]:

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

4.2 Examining the hypotheses

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

In relation to *Clustering Hypothesis*, we clustered each ranked list into 10 clusters using our clustering algorithm. Table 5 shows some statistical information for the clustering results. The first row lists four kinds of clusters containing no, 1, 2-10 and more than 10 relevant document(s). The second row shows the corresponding percentage of each kind of cluster.

The third row shows the percentage of relevant documents in each kind of cluster.

From Table 5, we can make two observations. First, about 50% of the clusters 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 the conclusion that relevant documents are concentrated in a few clusters.

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

Table 5. *Distribution of relevant docs.*

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

To test the *Fusion Hypothesis*, we computed $R_{overlap}$ and $N_{overlap}$ for each combination pair. 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 the *Fusion Hypothesis* holds for the retrieval results we obtained.

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

Table 6. *$R_{overlap}$ and $N_{overlap}$ values of combination pairs.*

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*

4.3 Comparison with conventional fusion methods

First, we studied three combination methods that were proposed by Fox, namely, CombMAX, CombSUM, and CombMNZ. Their fusion results for the same data set are listed in Table 7. The last row lists the average precision of each combination strategy. Since the average precision of the individual retrieval systems is 0.3086 (see Table 4), each of these three fusion methods has improved significantly in terms of the average precision. CombSUM appears to be the best one among them. This confirms the observation in [Fox 1994].

Then, we compared the performance of our approach with that of the other three methods, as shown in the last row in Table 7. Our new approach achieved 3% improvement over CombSUM. We also find that among all the 21 combination pairs, 17 of them are improved, compared to the results obtained using the CombSUM approach. We mark these rows with the character *.

Table 7. *Average precision of each combination pair.*

Combination pair	Comb MAX	Comb SUM	Comb MNZ	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*

Comparing the results shown in Table 7 with those listed in Table 6, we find that the pairs with a $R_{overlap}$ of over 0.80 correspond to better combination performance. We call this kind of pair a *combinable pair*. For example, BrklyCH1 & CLCHNA is a *combinable pair*. Although the average combination performance is 0.3654 (using our approach), almost all the *combinable pairs* exceed the average performance[3]. This again confirms the conclusion in both [Lee 1997] and [Vogt 1998] that the performance of fusion heavily depends on $R_{overlap}$. It also reveals the limitation of our approach and of other linear fusion techniques in that a high overlap of relevant documents is a pre-requisite for performance enhancement. For those pairs that don't satisfy this pre-requisite, normal fusion may even decrease retrieval performance.

We also compared our approach with the optimal linear combination. Since ranked lists

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

are combined linearly, only the ratio of the two weights affects the final performance:

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

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 was optimized using golden section search [Press 1992]. This approach was adopted in [Vogt 1998]. The average precision for the optimal linear combination we obtained is 0.3714. As shown in Fig.3, our approach performs better than CombSUM and CombMAX and is very close to CombBest.

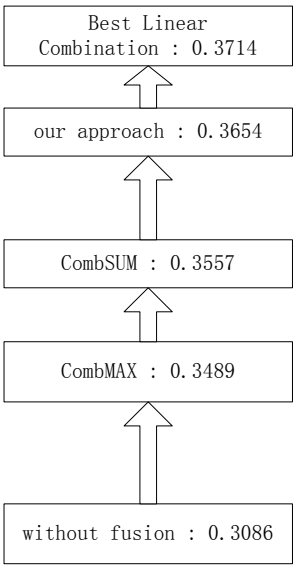

Figure 3 *Performance of different approaches.*

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

4.4 Impact of cluster size

We also studied the impact of cluster size. Table 8 shows the experimental results. When the cluster size varied from 200 to 5, the average precision did not change much. The maximum value was 0.3675 when the cluster size was 25 and the minimum value was 0.3621

when the cluster size was 200. This shows that the cluster size setting has very little impact in our approach.

Table 8. *Impact of cluster size.*

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

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

When the 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:

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

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

When the cluster size is set to 1, each document forms a cluster by itself. Those documents appearing in both ranked lists will be improved. For those documents that only appear in one ranked list, their relevance will remain 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 the CombSUM approach because this factor is close to 1.

The impact of the cluster size setting is illustrated in Fig.4. From this figure, we find that fusion combined with clustering is consistently better than the approaches that do not include clustering (where cluster size = 1000). We find that a setting size to 25 gives the best combination when the ranked list has a size of 1,000.

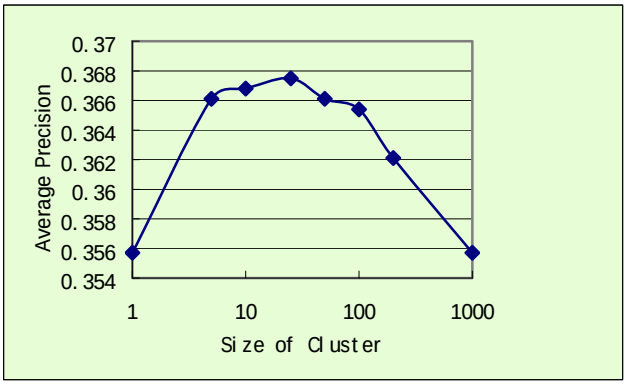

Figure 4 *Impact of cluster size.*

5. Conclusion

Combining multiple retrieval results is certainly a practical technique for improving the overall performance of information retrieval systems. In this paper, we have proposed a novel fusion method that can be combined with document clustering to improve retrieval performance. 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 using our re-ranking approach. Finally, conventional fusion is carried out to produce an adjusted ranked list.

Since our approach is based on two hypotheses, we first verified them by means of experiments. We also compared our approach with other conventional approaches. The results show that each of them achieves some improvement, and that our approach compares favorably with them. We also investigated the impact of cluster size. We found that our approach is rather stable under variation in the size of clusters.

Although our method showed good performance in our experiments, we believe it still can be improved further. A better clustering algorithm for identifying more reliable clusters and more elaborate formula for re-ranking ranked lists should lead to further improvement. These will be topics for our future work.

References

Bartell,B.T., Cottrell,G.W., and Belew,R.K., "Automatic Combination of Multiple Ranked Retrieval Systems," *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 173-181.

D.R.Cutting, D.R.Karger, J.O.Pedersen, and J.W.Tukey, "Scatter/gather: A Cluster-based Approach to Browsing Large Document Collections," *Proceedings of the 15th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1992, pp. 126-135.

Fox,E. and Shaw,J., "Combination of Multiple Searches," The Second Text Retrieval Conference (TREC2), NIST Special Publication 500-215, 1994, pp. 243-252.

Hearst,M.A., and Pedersen,J.O., "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results," *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1996, pp. 76-82.

J.H.Lee. "Analyses of Multiple Evidence Combination.," Proceedings of the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1997, pp. 267-276.

A.Leuski and J.Allan, "The Best of Both Worlds: Combining Ranked List and Clustering," *CIIR Technical Report IR-172*, 1999, http://cobar.cs.umass.edu/pubfiles/ir-172.ps.

A.Leuski and J.Allan, "Improving Interactive Retrieval by Combining Ranked List and Clustering," Proceedings of RIAO(Recherche d'Informations Assistee par Ordinateur = Computer-Assisted Information Retrieval) 2000 Conference, 2000, pp. 665-681.

C.J.van Rijsbergen, *Information Retrieval*, Butterworths, London, second edition, 1979.

Thompson,P., "A Combination of Expert Opinion Approach to Probabilistic Information Retrieval, part I: The Conceptual Model," *Information Processing and Management*, 26(3) 1990, pp. 371-382.

Vogt,C., Cottrell,G., Belew,R. and Bartell,B., "Using Relevance to Train a Linear Mixture of Experts," *Proceedings of the 5th Text Retrieval Conference (TREC5), NIST Special Publication 500-238*, 1997, pp. 503-516.

Vogt,C. and G.Cottrell., "Predicting the Performance of Linearly Combined IR Systems," Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1998, pp. 190-196.

Vogt,C. and Cottrell,G., "Fusion Via a Linear Combination of Scores," *Information Retrieval*, 1(2-3), 1999, pp. 151-173.

E.Voorhees, D.Harman, "Overview of the Sixth Text Retrieval Conference (TREC-6)," *NIST Special Publication 500-240*, 1997. pp. 1-24.

Press,W.H., Teukolsky,S.A., Vetterling,W.T., and Flannery,B.P., *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, 1992.