BJTU TRECVID 2007 Video Search

Shikui Wei, Yao Zhao, Zhenfeng Zhu, Nan Liu , Yufeng Zhao, Fang Wang, Xie Lin
Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China
E-mail: shkwei@gmail.com

ABSTRACT

In this paper, we describe our experiments of search task for TRECVID 2007. This year we participated in the automatic video search subtask, and submitted six runs with different combination of approaches to NIST. Using the text only based search engine used in last year, the run F_A_1_JTU_FA_1_1 provides a baseline search result list. In order to bring up true relevant results, a multi-view based reranking method is employed for reordering the search results derived from run F_A_1_JTU_FA_1_1. Specifically, initial search results, which are represented by multiple distinct feature views, are first divided into several clusters on individual feature views. According to their relevance to query intention, clusters on each feature view are mapped into predefined ranks. With these ranked clusters on all feature views, a strikingly new Cross-Reference (CR) method are employed to fuse them into a unified result ranking. The following five runs test the effect on reranking performance of different combination of clustering methods and fusion strategies. F_A_1_JTU_FA_1_2: NCut Clustering + Late-Fusion. F_A_1_JTU_FA_1_3: NCut Clustering + Single View A. F_A_1_JTU_FA_1_4: NCut Clustering + Single View B. F_A_1_JTU_FA_1_5: NCut Clustering + Bi-Fusion. F_A_1_JTU_FA_1_6: NCut Clustering + Single View A+Single View B.

1. INTRODUCTION

Considering that the final aim of search engine is to meet user's information needs, it is reasonable to take into account factors such as user satisfaction and user behavior on returned results. Usually, for a real world search scenario, users are rarely patient to go through all results returned from search engine in order to find information needed. Instead, they often only check the top-ranked documents among the returned results [5]. Analysis on click-through data from a very large web search engine log also shows that users are usually interested in a very small set of relevant shots in top-ranked results [4]. Hence, it is more crucial to give higher precision on the top-returned results so as to meet user's need instantly.

As a kind of effective solutions, reranking techniques [7,8] have been successfully applied in the text-based web search field. The goal of reranking techniques is to reorder the similarity-based retrieval results and to bring up true relevant documents. Intuitively, it is a good idea to extend this kind of method to the video search field. While some efforts have been made for this issue, it attracted relatively little attention in multimedia retrieval community. Although aforementioned reranking schemes improve average precision of video search to some extent, they pay little attention to accuracy of the top of result list, that is, all shots in initial result list are treated equally during the phase of reordering. To have more high precision on top-ranked results, a multi-view based scheme is proposed for reordering top N initial results. The fundamental idea is that video search can be benefited from clustering initial results separately on multiple independent feature views at a relatively low noise level. However, the idea of multi-view [9-11] has been suggested originally for the purpose of cooperative training in a semi-supervised fashion. Multi-view method, here, is utilized to bring up the most relevant results in the initial result list, which is different from its original role. Our approach contains three main stages: clustering initial results on each feature view, determining the rank of each cluster, and fusing ranked clusters into a new result ranking. For the first issue, an advanced Normalized Cuts (NCuts) clustering algorithm [13,14] is employed to cluster initial results. The experimental results show that NCuts clustering algorithm used in our scheme outperforms some other general clustering ones such as k-means. Subsequently, clusters obtained on each feature view are mapped into predefined ranks in term of their relevance to query intention. Finally, a novel fusion method taking cross-reference strategy as core is presented for combining these ranked clusters into a unified ranking list. Experimental results indicate that the proposed reranking method indeed gives higher accuracy on the most top of result list.

The rest of this paper is organized as follows. In section 2, the proposed scheme is then described. Section 3 gives performance analysis of submitted runs. Conclusions are drawn in section 4.

2. MULTI-VIEW RERANKING SCHEME

The common goal of various reranking methods is to reorder initial result list and bring effectively up true relevant shots. However, improving the accuracy of the most top-ranked results is more crucial than increasing the average precision at a great depth as analyzed previously. While previous works have presented some working algorithms, most of them deal with initial results in a single feature view; that is, shot representations from multiple modalities are concatenated into a single feature. As a result, initial results at different ranks can only be treated impartially, and less attention is focused on accuracy of the most top-ranked results which users are most interested in. In our scheme, multiple modality representation for shot is explicitly split into several approximately independent feature views, so each feature view give a complete representation for initial results. After clustering initial results and ranking these clusters in each single feature space, each shot is favorably handled with our novel cross-reference method across different feature views.

In this section, we will give a detailed description about our proposed reranking scheme, which includes multi-view clustering strategy, a new method of ranking clusters on each feature view, and a novel method of fusing multi-view clusters. The overview of proposed reranking scheme is illustrated in Fig 1. Note that only two feature views are considered here, it can be easily extended to much more approximately independent feature views.

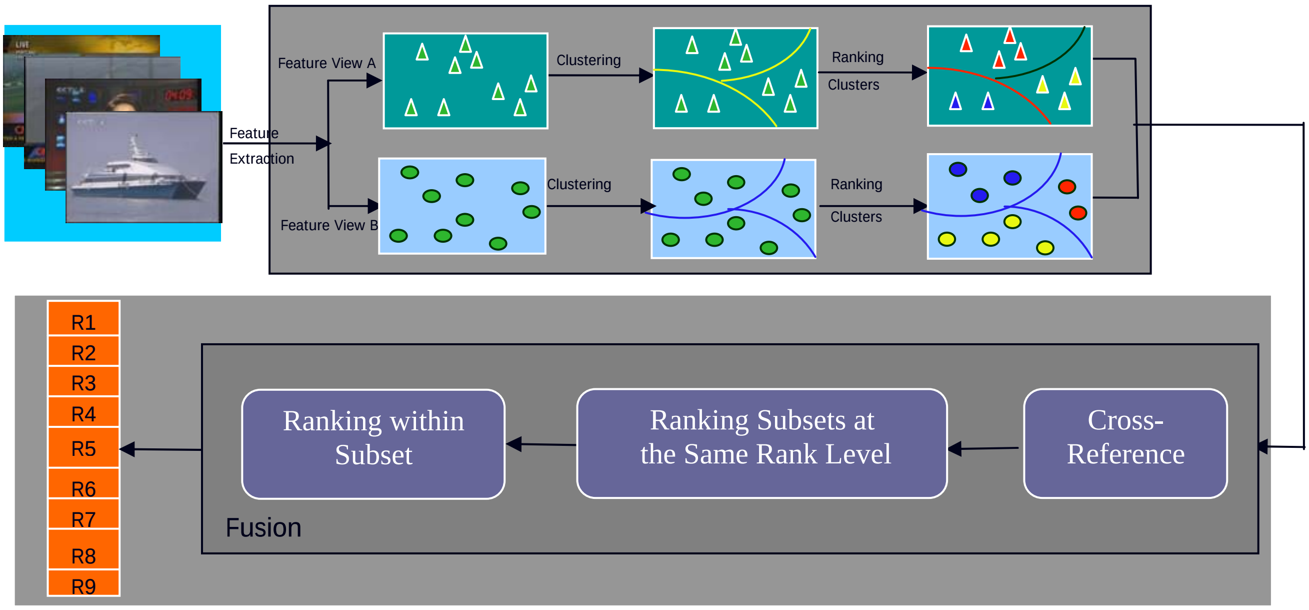


Fig.1 Overview of the proposed re-ranking scheme. The upper box shows how to generate ranked clusters in individual feature views, and the lower box illustrates how to fuse ranked clusters from two feature views. Note that red, yellow and blue indicate the high, median and low ranks, respectively.

2.1. Multi-View Clustering Strategy

The essential of multi-view strategy is that each example is explicitly represented by multiple distinct feature views; hence, the improvement of learning quality based on each feature view will be benefited from other independent views iteratively, and a more reliable learning outcome can be expected finally. Multi-view strategy has been successfully applied to concept detection domain [11], in which the presence of multiple distinct feature views of each example is utilized to leverage unlabeled data by training a separate learner on each feature view. In our case, however, with characterizing the shots from initial search on two distinct feature views, some clustering algorithm can be separately employed to partition those shots into a fixed number of clusters on each feature space due to some certain distance measure. While our reranking scheme is not dependent on the specific clustering method in principle, clustering performance indeed has a slight impact on the reranking quality. As one of the most popular clustering algorithm, k-means clustering can not partition points that are non-linearly separable in input space. Instead, spectral clustering method can deal with this problem to a certain extent. The NCuts clustering algorithm, which minimizes the normalized cuts of a graph, is one of popular spectral clustering methods. Specifically, this method places its optimization criterion on both total dissimilarity among different clusters and total similarity within clusters [14] and turns out to be suitable for partitioning initial results in feature space. Given a fixed N shots, feature representation of each shot is considered as a vertex of set V, and set E consists of N by N edges with a weight matrix W of edges. Note that there is a connection between any two vertices in set E, and the weight of edge is computed as a certain similarity measure between two vertices. After building an undirected graph $G=(V,E,W)$, the purpose of NCuts clustering is to partition set V into several disjoint sets $V_1,V_2,\cdots,V_k$.

By this way, we can obtain a certain number of clusters on each feature view, which paves the way for implementation of our cross-reference ranking algorithm. In order to avoid partitioning out too small set of points, the number of clusters is fixed to be three.

2.2. Cluster Ranking

As aforementioned，an important step in our scheme is to correctly assign a rank for each cluster so as to implement cross-reference operation. In this section, an effective cluster ranking approach is proposed to meet our requirement. In fact, our proposed approach is partly inspired by Pseudo-relevance feedback (PRF) method, which can automatically modify query model and refine retrieval results effectively, as opposed to traditional relevance feedback (RF) method with explicit user intervention. PRF is based mainly on a questionable assumption that top-ranked documents obtained from initial search have a significant fraction being relevant. Usually, top-ranked documents are treated directly as true relevant results for updating query model and refining search results. Therefore, the effectiveness of PRF is dependent strongly on the precision of initial search. Unfortunately，current video search engine can not meet this assumption well due to the poor accuracy and scarceness of true relevant shots in the top-ranked results. Nevertheless, R.Yan et al.[3] propose a working reranking algorithm by automatically conducting feedback process on different modalities despite the drawback of PRF. However, the effectiveness of this method will fall down with the number of relevant shots decreasing in the most top-ranked documents. To avoid it, W. H. Hsu et al. [2] utilize estimated “soft” pseudo-labels to learn user preference, instead of “hard” ones. Although this fashion can solve this issue to some extent, all three pseudo-labeling strategies used in [2] are based only on the ranks of initial search; that is, the effectiveness is still strongly dependent on the performance of initial search.

Browsing over the most top-ranked results derived from automatic search results submitted to TRECVID 2006, we observed that the true relevant shots are of much similarity in perception, whereas dissimilarity spreads over irrelevant ones, so we name them by centralization attribute of relevant shots and decentralization property of irrelevant ones, respectively. Instead of treating the top-ranked results as true positive examples, in our approach, we utilize implicitly the labels of top-ranked results by exploring the centralization and decentralization properties of the initial search results. However, similar to [2], the most top-ranked initial results are still considered the most informative instances containing the query intention of users, and are partly used for ranking clusters. As a consequence, the problem of assigning ranks to clusters is equivalent to similarity measure between the set of the most top-ranked results and clusters. Hausdorff distance [12], which is the maximum distance of a set to the nearest point in the other set, defines a distance measure between two sets. More formally, Hausdorff distance between set A and set B can be formulized as follows:

$$HD(A,B) = \max\{hd(A,B),\ hd(B,A)\} \qquad (1)$$

$$hd(A,B) = \max_{a\in A}\{\min_{b\in B}\{d(a,b)\}\} \qquad (2)$$

$$hd(B,A) = \max_{b\in B}\{\min_{a\in A}\{d(b,a)\}\} \qquad (3)$$

where, a and b are elements (feature points) in set A and set B, respectively, and d(a,b) can be any distance metric between point a and b. According to the definition of Hausdorff distance, it measures the similarity (more properly, dissimilarity) of set A and set B by taking into account all points in two sets. Consequently, this distance metric does not work well when a lot of noisy points exist. In our case, only top-30 initial results, namely, the most top ranked results, are taken as set A, and irrelevant examples in set A can be considered as the noisy points, as opposed to true relevant results. From the analysis of section 2, a lot of irrelevant results indeed exist in the top-30 results. Therefore, Hausdorff distance can not perfectly meet our need for ranking clusters.

In contrast, Partial Hausdorff distance metric[12], which can automatically select some best matching points to measure similarity of two sets, is more suitable for measuring two noisy sets. The definition of partial Hausdorff distance between set A and set B is given as:

$$PHD(A,B) = \max\{phd(A,B),\ p\,hd(B,A)\} \qquad (4)$$

$$phd(A,B) = K^{th}_{a\in A}\{\min_{b\in B}\{d(a,b)\} \qquad (5)$$

$$phd(B,A) = K^{th}_{b\in B}\{\min_{a\in A}\{d(b,a)\} \qquad (6)$$

where, $K^{th}_{a\in A}$ denotes the K^{th} value of distance list(from set A to set B) in ascending order, and $K^{th}_{b\in A}$ denotes the K^{th} value of distance list (from set B to set A) in ascending order.

However, in our case, only partial directed Hausdorff distance phd(A,B) is utilized to measure two noisy sets, where A denotes the set of top-30 results, B is one of clusters. Using this distance, the centralization and decentralization properties can be suitably considered by selecting a proper value of K. In our experiments, the value K is selected against statistical amount of true relevant results in top-30 results over 76 search runs submitted to TRECVID 2006[1, 15] and is fixed to 5.

Using the partial directed Hausdorff distance, for each cluster, the distance to the set of top-30 results is computed, and those clusters are ranked by their respective values of this distance.

2.3. Fusion Strategy

With previous preprocessing steps, the initial results are separately partitioned into three distinct clusters on each feature view, and then these clusters are mapped into predefined ranks. However, our final goal is to obtain a unified and improved ranking of initial results. Therefore, a key issue is how to fuse ranked clusters from two approximately independent feature views into a unified and improved result list, taking into account the intention of paying much more attention to accuracy of the most top-ranked results. In the following part, we will give a detailed depiction about our fusion strategy, which includes three main components: cross-reference combination, ranking fused subsets and ranking within each subset.

As the core of fusion strategy, cross-reference method plays a key role in approaching our goal. The fundamental idea of cross-reference is that ranked clusters from two feature views are combined into a unified subset list by intersecting each cluster on feature view A with clusters on feature view B in descending order. That means that shots in high ranked cluster on both feature view A and B are given a higher score so that the higher accuracy is given to the most top-ranked subset of list. The criteria, or named by Cross-reference, is formulized as:

$$Rank(A_i) \succ Rank(A_m), \quad if\ i \prec m, \quad i,m = 1,2,3 \qquad (7)$$

$$Rank(B_j) \succ Rank(B_n) \quad if\ \ j \prec n, \quad j,n = 1,2,3 \qquad (8)$$

$$\begin{aligned} &Rank(A_i \cap B_j) \succ Rank(A_m \cap B_n), \\ &\quad if\ (i+j) \prec (m+n),\ i, j, m, n = 1,2,3 \end{aligned} \qquad (9)$$

here, A_i is the i^{th} cluster on feature view A; B_j denotes the j^{th} cluster on feature view B; $A_i \cap B_j$ stands for the intersection of cluster A_i and B_j; expression $Rank(A_i) \succ Rank(A_m)$ and $Rank(B_j) \succ Rank(B_n)$ mean that the clusters on both feature view A and B are ranked from high to low in ascending order of subscripts.

As a matter of fact, the ranks of subsets can not be determined using merely those criteria above when (i+j) = (m+n), such as subset ($A_1 \cap B_2$) and ($A_2 \cap B_1$). To address this issue, those subsets being of the same rank level are ordered using the same method discussed in subsection 2.2. This rule can be followed as:

$$\begin{gathered}Rank(A_i \cap B_j) \succ Rank(A_m \cap B_n),\\ if\ \ (i+j)=(m+n),\ phd(A, A_i \cap B_j) \prec phd(A, A_m \cap B)\end{gathered} \qquad (10)$$

where, A is the set of top-30 initial results. Note that it is enough to carry out this ranking process on a single feature view, which is alternative.

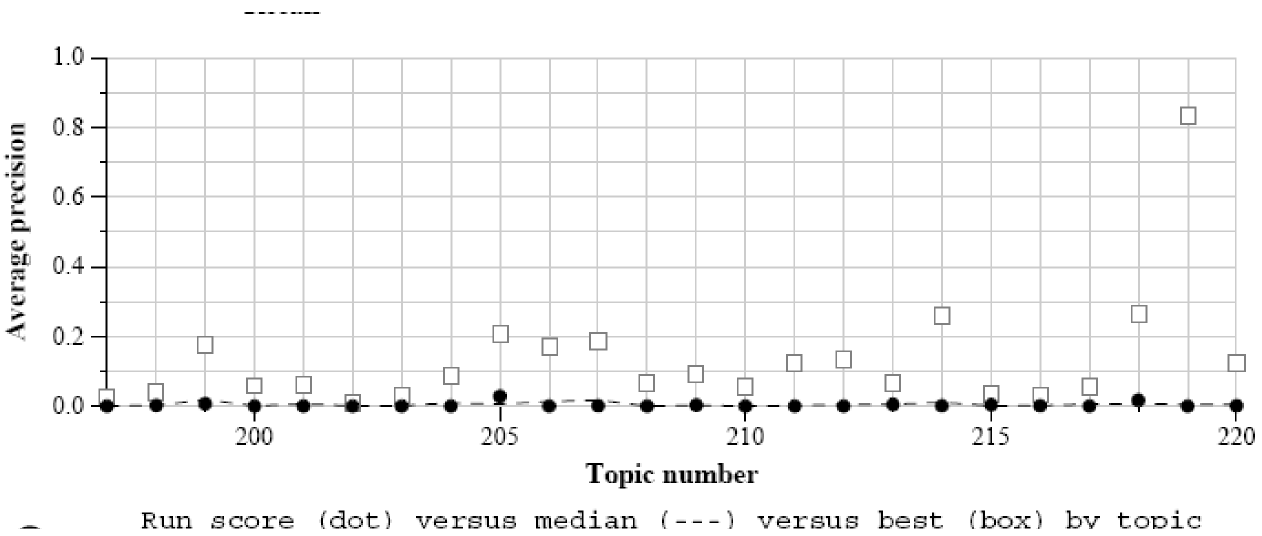
So far, using the criteria above, the initial search results are mapped into several ordered subsets, and then an ordered subset list $U = U_1 \cup U_2 \cdots \cup U_9$ is formed. Although the ranks of shots in different subsets can be compared against the ranks of corresponding subsets, we don’t know which shot within the same subset is more relevant to query. Hence, the final step of proposed reranking scheme is to reorder the results within the same subset. However, as the informative indications, the ranks of initial search are rarely explored through the whole reranking scheme except for cluster ranking. Here, the initial information is used again to reorder shots in the same subset, that is, those shots within subset are ordered according to their text retrieval scores or ranking orders. This rule is defined as follow:

$$\begin{gathered}Rank(S_{i,m}) \succ Rank(S_{i,n}),\\ if\ TextScore(S_{i,m}) \succ TextScore(S_{i,n}), S_{i,m}, S_{i,n} \in U\end{gathered} \qquad (11)$$

where, $s_{i,m}$ and $s_{i,n}$ are shots in subset U_i, $TextScore(S_{i,m})$ and $TextScore(S_{i,n})$ are the initial search ranks.

3. EXPERIMENTS

Before carrying out reranking scheme, an initial ranking must be obtained. In our experiments, a fully automatic text-based video search engine is constructed to give a rough search ranking against only text information. Actually, the search engine is the same as one used in TRECVID 2006. The more detailed information can refer to our prior work [16]. Using this search engine, we obtain an initial search list of total 2000 shots for each query topic, ie. F_A_1_JTU_FA_1_1, and our proposed reranking scheme is based on this search. The performance of initial search is illuminated in Fig.2. As shown in Fig.2, the search results are not encouraging due to the lock of informative text content in TRECVID 2007 video corpus. In fact, our reranking scheme is on the basic of the assumption that there is a quite few of relevant shots in initial search results. Hence, all of our reranking schemes are useless when this assumption is demolished, just like F_A_1_JTU_FA_1_1 run. From the returned evaluation results, we also validate the conclusion. This is, the results reranked do not bring up the true relevant shots. Actually, we also test our reranking scheme in TRECVID 2006 video corpus which is of more informative text content, the results are encouraging, this is, the reranking scheme indeed bring up true relevant shots, especially improving the precision in the most top returned results.



4. CONCLUSIONS

In this paper, a novel multi-view based reranking algorithm is presented for reordering initial result list derived from some video search engines so as to bring up true relevant shots. Specifically, each shot in initial results is represented by multiple distinct feature views separately. And then initial results are partitioned into several clusters on individual feature views using an advanced NCuts clustering algorithm. According to their relevance to query intention, clusters on each feature view are mapped into predefined ranks. Using a novel fusion strategy with Cross-Reference method, these ranked clusters are fused into a unified result list. The main merit of the proposed reranking technique is to pay much more attention on the accuracy of the most top-ranked results.

REFERENCES

[1] A. Smeaton and T. Ianeva, "TRECVID-2006 Search Task," presented at TREC Video Retrieval Evaluation Online Proceedings, Gaithersburg, USA, 2006.

[2] W. H. Hsu, L. S. Kennedy, et al., “Video Search Reranking via Information Bottleneck Principle,” In 14th annual ACM international conference on Multimedia, Santa Barbara, CA, USA, pp. 35-44, 2006.

[3] R. Yan and A. G. Hauptmann, "Co-retrieval: a boosted reranking approach for video retrieval," Vision, Image and Signal Processing, IEE Proceedings, vol. 152, pp. 888-895, 2005.

[4] T. Joachims, "Optimizing search engines using clickthrough data," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 133-142, 2002.

[5] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, "Adapting ranking SVM to document retrieval," presented at the 29th annual international ACM SIGIR conference on Research and Development in Information Retrieval, Seattle, Washington, USA, 2006.

[6] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a very large web search engine query log," ACM SIGIR Forum, vol. 33, pp. 6-12, 1999.

[7] T. H. Haveliwala, "Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search," Knowledge and Data Engineering, IEEE Transactions on, vol. 15, pp. 784-796, 2003.

[8] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," Journal of the ACM (JACM), vol. 46, pp. 604-632, 1999.

[9] A. Blum, T. Mitchell, “Combining Labeled and Unlabeled Data with Co-Training,” In Proceedings of the Workshop on Computational Learning Theory, ACM, New York, USA, pp. 92-100, 1998.

[10] K. Nigam, R. Ghani, “Understanding the Behavior of Co-training,” In Proceedings of the Workshop on Text Mining, ACM, 2000.

[11] R. Yan, M. Naphade, “Multi-Modal Video Concept Extraction Using Co-Training,” In International Conference on Multimedia and Expo, IEEE, pp. 514-517, 2005.

[12] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 15, pp. 850-863, 1993.

[13] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 551-556, 2004.

[14] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888-905, 2000.

[15] TRECVID. TREC Video Retrieval Evaluation. In http://www-nlpir.nist.gov/projects/t01v

[16] S. K. Wei et al., “BJTU TRECVID 2006 Video Retrieval System,” In TREC Video Retrieval Evaluation Online Proceedings, TRECVID, Gaithersburg, USA, 2006.