Co-Retrieval: A Boosted Reranking Approach for Video Retrieval

Rong Yan and Alexander G. Hauptmann

School of Computer Science
Carnegie Mellon University
Pittsburgh PA, 15213, USA
{yanrong, alex+}@cs.cmu.edu

Abstract. Video retrieval compares multimedia queries to a video collection in multiple dimensions and combines all the retrieval scores into a final ranking. Although text are the most reliable feature for video retrieval, features from other modalities can provide complementary information. This paper presents a reranking framework for video retrieval to augment retrieval based on text features with other evidence. We also propose a boosted reranking algorithm called Co-Retrieval, which combines a boosting type algorithm and a noisy label prediction scheme to automatically select the most useful weak hypotheses for different queries. The proposed approach is evaluated with queries and video from the 65-hour test collection of the 2003 NIST TRECVID evaluation. [1]

1 Introduction

The task of content-based video retrieval is to search a large amount of video for clips relevant to an information need expressed in form of multimodal queries. The queries may consist merely of text or also contain images, audio or video clips that must be matched against the video clips in the collection. Specifically this paper focuses on the content-based queries which attempt to find semantic contents in the video collection such as specific people, objects and events. To find relevant clips for content-based queries, our video retrieval system needs to go through the following steps as indicated in Figure 1. First, various sets of index features are extracted from the video library through analysis of multimedia sources. Each video clip (or shot) is then associated with a vector of individual retrieval scores (or ranking features) from different search modules, indicating the similarity of this clip to a specific aspect of the query. Finally, these individual retrieval scores are fused via a weighted linear aggregation algorithm to produce an overall ranked list of video clips.

It is of great interest to study the combination methods in the final step. This work considers approaches which rerank the retrieval output originally obtained from text features, using additional weak hypotheses generated from other

[1] This research is partially supported by Advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037 and MDA904-02-C-0451.

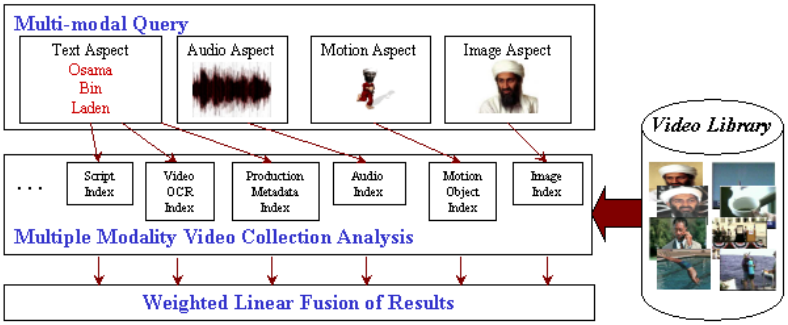

Fig. 1. Overview of our video retrieval system

modalities as evidence. Text retrieval first finds a set of relevant shots for each query, with associated scores that define an initial ranking. The selected weak hypotheses are then weighted and linearly combined in an attempt to improve upon the initial ranking.

Merely combining weak hypotheses with fixed weights or asking users to explicitly set weights are either inflexible or unrealistic. It is desired for the system to learn the linear weights automatically. Put in another way, the system should be able to pick out the most related concepts without feedback from users. For example, for the query "finding people on the beach" the ideal system can choose outdoors and people features to improve the initial retrieval. To achieve this, we apply a boosting type algorithm, which repeatedly learns weak classifiers from a reweighted distribution of the training data and combines the weak classifiers into one composite classifier. We also provide a noisy label prediction scheme which allows it to improve the initial ranking without any external training data. Experiments applied the proposed approach to a video test collection of over 65 hours from the 2003 content-based video retrieval track [8].

2 A Reranking Framework for Video Retrieval

Based on evidence from the best-performing video retrieval systems in the 2001 and 2002 NIST TREC Video Retrieval evaluation tasks, text features are demonstrated to be the most reliable ranking features for selecting semantically relevant shots in video retrieval. Text features span several dimensions including automatic speech recognition(ASR), closed captions(CC), video optical character recognition(VOCR) and production metadata such as published descriptions of the video. Typically a number of nearby shots are also retrieved since temporal proximity can somehow indicate content closeness. However, ranking of video shots cannot simply rely on these text features. One important reason is that words may be spoken in the transcript when no associated images are present, e.g. a news anchor might discuss a topic for which no video clips are available. A reporter may also speak about a topic with the relevant footage following later in the story, resulting in a major time offset between the word and the relevant video clips. As shown in figure 2(a), text retrieval will at times assign high scores to shots of studio settings or anchors which is usually not desirable. Moreover,

Fig. 2. The key frames of top 8 retrieved shots for query "Finding Tomb at Arlington National Cemetery". These are retrieval results based on (a) text features (b) image features (c) text plus image features while filtering out news anchors and commercials

word sense ambiguity may result in videos retrieved of the wrong meanings, e.g. either a river shore or a financial institution is possible for the word 'bank'. Speech recognition errors or VOCR errors may also result in incorrect retrieval. In general, retrieval using text features exhibits satisfactory recall but relatively low precision.

Fortunately, many complementary sources from various video modalities can be used to rerank the text retrieval output. These sources includes audio features, motion vectors, visual features (e.g. color, edge and texture histograms), and any number of pre-defined high-level semantic features (e.g. a face detector and an anchor detector). Generally speaking, none of these can fully capture the full content of the shots and therefore retrieval results based only on these features are mostly unsatisfying. Figure 2(b) depicts the top results of image-only retrieval which returns nothing related to the query. However, these weak features can provide some indication of how closely the video shots are related to the given specific query examples. They can also filter out irrelevant shots such as anchorpersons or commercials. Figure 2(c) illustrates the advantage of weak features which can augment text retrieval by finding the similar objects and filtering news anchor plus commercial shots.

These observations indicate that we should rely on text-based retrieval as the major source for answering semantic queries, while using the weak ranking functions from other modalities in combination to refine the ranking from the text retrieval. To be more general in the implementation, we convert the weak ranking features into a set of [-1,1]-valued weak hypotheses. Therefore, we propose the following re-ranking framework for video retrieval,

$$F(\mathbf{x_i}, \lambda) = \lambda_0 F_0(\mathbf{x_i}) + \sum_{t=1}^{m} \lambda_t h_t(\mathbf{x_i}) \tag{1}$$

where λ_t is the weight for the t^{th} ranking function, $\mathbf{x_i}$ is the i^{th} video shot, $F_0(\cdot)$ is the base ranking function generated by text retrieval and $h_t(\cdot)$ is the output of the t^{th} weak hypothesis. Without loss of generality, we assume $F_0(\mathbf{x_i}) = 1$ when shot $\mathbf{x_i}$ is found to be relevant by text retrieval, otherwise $F_0(\mathbf{x_i}) = 0$. λ_0 is

typically set to be $m \max_i(\lambda_i)$ after λ_i are learned. This allows F_0 to dominate the retrieval results while the other weaker hypotheses h_t re-rank and adjust the output provided by F_0.

3 Co-Retrieval: Boosted Reranking for Video Retrieval

In this section, we propose the Co-Retrieval algorithm which combines a boosting-type learning approach and a noisy label prediction scheme to estimate the weights λ without any external training data or user feedback.

3.1 Boosted Reranking with Training Data

Let us begin with considering the case when a number of training data $\{x_i, y_i\}$ are available for each query, where $y_i \in \{-1, +1\}, i = 1..N$. The goal of a learning algorithm is to find a setting for λ which leads to better retrieval outputs than F_0. Assuming all of x_i can be found by text retrieval, i.e. $F_0(\mathbf{x_i}) = 1$, we only need to optimize $F(\mathbf{x_i}, \lambda) = \sum_{t=1}^{m} \lambda_t h_t(\mathbf{x_i})$. In the current setting, the learning algorithm will produce a function $H : X \to \mathbf{R}$ which approximates the ordering encoded by the training data $\{x_i, y_i\}$. Formally, the boosting algorithm can be designed to minimize a loss function related to ranking misordering, i.e. $\lambda = \arg\min_\lambda Loss(\{y_i, F(x_i, \lambda)\})$. In analogy to classification problems, the loss function can be set to a monotonically decreasing function of margin $y_i F(x_i, \lambda)$, i.e. $\sum_{i=1}^{N} L\left(y_i F(x_i, \lambda)\right)$. Two typical choices for function L are the exponential loss $\exp(-x)$ used by AdaBoost and the logit loss $\log(1 + \exp(-x))$ used by logistic regression. However, it has been argued that it is more reasonable to optimize ranking misordering through relative preferences rather than using an absolute labeling. Along this direction, the RankBoost algorithm proposed by Freund et al. [2] develops a boosting framework with ranking loss for combining multiple weak ranking functions. As a special case when the feedback function is *bipartite*, they provide an efficient implementation which actually minimizes the following ranking loss function, i.e. $\sum_{y_i=+1} \sum_{y_j=-1} e^{-F(x_i,\lambda)+F(x_j,\lambda)}$.

To handle different loss functions, we apply a unified boosting type algorithm for learning the combination of weak hypotheses. The boosting algorithm shown in Figure 3 are modified from the parallel update algorithm proposed by Collins et al.[3]. For each round k, the algorithm first updates the distribution $q_{k,i}$ in a manner that increases the weights of examples which are misclassified. We will consider three loss functions in this step, i.e. exponential loss, logit loss and ranking loss. They have the following update rules respectively,

$$q_{k+1,i} = \begin{cases} q_{k,i} \exp\left(-\sum_{j=1}^{n} \delta_{k,j} M_{i,j}\right) \\ q_{k,i} \left[(1 - q_{t,i}) \exp\left(-\sum_{j=1}^{n} \delta_{k,j} M_{i,j}\right) + q_{k,i}\right]^{-1} \\ q_{k,i} \exp\left(-\sum_{j=1}^{n} \delta_{k,j} M_{i,j}\right) \sum_{y_i \neq y_l} q_{k,l} \exp\left(-\sum_{j=1}^{n} \delta_{k,j} M_{l,j}\right) \end{cases} \tag{2}$$

A new step, i.e. step 2, is added to balance the distribution between positive and negative examples. In fact many boosting algorithms take some reweighting or filtering approaches to obtain a balanced distribution such as the RankBoost.B[2],

Input: Matrix $\mathbf{M} \in [-1,1]^{m \times n}$ where $M_{ij} = y_i h_j(x_i)$ and $\sum_{j=1}^{n} |M_{ij}| \leq 1$ for all i. N^+ is the number of positive examples and N^- is the number of negative examples
Output: $F(\mathbf{x_i}, \lambda) = \sum_{t=1}^{m} \lambda_t h_t(\mathbf{x_i})$, where λ_1,..., λ_m optimize $Loss(\{y, F(x)\})$.
Algorithm:
Let $\lambda_1 = 0$, $q_0 = (0.5, 0.5, ...)$
For $k = 1, 2, ...$

1. Compute distribution q_k given $\mathbf{M}$, δ_k and q_{k-1}
2. For every positive example $\mathbf{x_i}$, balance the distribution $q_{k,i} = N^- q_{k,i} / N^+$
3. For $j = 1, .., m$:
   $W_{k,j}^+ = \sum_{i:sign(M_i j)=+1} q_{k,i} |M_{ij}|$
   $W_{k,j}^- = \sum_{i:sign(M_i j)=-1} q_{k,i} |M_{ij}|$
   $\delta_{k,j} = \frac{1}{2} \log \left(W_{k,j}^+ / W_{k,j}^- \right)$
4. Update parameter: $\lambda_{\mathbf{k+1}} = \lambda_{\mathbf{k}} + \delta_k$

Fig. 3. A unified boosting type algorithm with parallel-update optimization

otherwise a constant -1 hypothesis is the likely output from all weak hypotheses. Finally the update vector is computed as shown in step 3 and added to the parameters $\lambda_{\mathbf{k}}$. More details and the convergence proof can be found in [3].

The boosting algorithm requires access to a set of weak hypotheses $h(\cdot)$ produced from ranking features. The most obvious choice for weak hypotheses is equal to the normalized ranking features f_i, i.e., $h(x) = a f_i(x) + b$ where a, b are constants to normalize $h(x)$ to the range of $[-1, +1]$. If we only consider the relative ranking provided by the weak features instead of their absolute values, we can use the {-1,1}-valued weak hypotheses h of the form, i.e. $h(x) = 1$ if $f_i(x) > \theta$ otherwise $h(x) = -1$, where $\theta \in \mathbf{R}$ is some predefined threshold. In our implementation, we use the first definition for the features which compute the distance between given image/audio examples and video shots in the collection, e.g. the Euclidean distance from a color histogram. For the features generated by semantic detectors such as face and outdoors detectors, we choose the second definition because their relative ordering makes more sense than an absolute value. Rather than learning the threshold θ automatically, we prefer to fix the threshold to 0.5 in terms of posterior probability.

3.2 Learning with Noisy Labels

So far we assume training data are available for the boosting algorithm, however, collecting training data for every possible query topic on the fly is not feasible in general. Alternative approaches have to be developed to generate a reasonable weight assignment without requiring a large human effort to collect training data. Formally, considering the entire set of shots returned by text retrieval $\{x_1, ..., x_i, ..., x_n\}$, we need to assign a set of (noisy) labels y_i which allows the boosting algorithm to improve the retrieval performance.

Without loss of generality, let us assume $\{x_1, ...x_n\}$ are sorted in descending order of text retrieval scores and denote r_k the number of relevant shots in $\{x_1, ...x_k\}$. By analyzing the characteristics of text retrieval, we make the fol-

Top shots k/n%	15%	25%	50%	75%	100%
r_k	184(32.11%)	248(43.8%)	367(64.1%)	487(84.7%)	573(100%)

Table 1. The number of relevant shots r_k in top $k/n\%$ shots returned by text retrieval which is averaged over 25 TREC03 queries. The number in () is $(r_k/r_n) * 100\%$

lowing assumption in the rest of this paper: The proportion of relevant shots in $\{x_1, ...x_k\}$ is higher than in the entire set, i.e. $r_k/k \geq r_n/n$. In other words, the relevant shots are more likely to be higher ranked in the text retrieval. One partial explanation is that shots farther away from the content keyword location are lower ranked, with a lower probability of representing relevant concepts. Table 1 provides more insights to support our assumption. Therefore we can simply assign $\{y_1, ...y_k\}$ as +1 and $\{y_{k+1}, ..., y_N\}$ as -1. In practice, we can augment the raw text retrieval scores with some highly accurate features to improve noisy label prediction, e.g. use anchor detectors to filter out irrelevant shots.

However, because automatically generated training data is quite noisy, regularization is generally required to reduce the effect of overfitting. Instead of introducing a penalty function into the loss function, we suggest two types of regularization approaches, 1. Use a χ^2 test to select features with confidence interval 0.1; 2. Set λ_t to be 0 if $\lambda_t < 0$ for the nearest-neighbor-type features.

3.3 Related Work

Our approach builds on previous work which investigated the use of learning algorithms to improve ranking or retrieval. Collins et al. [4] considered a similar discriminative reranking approach to improve upon the initial ranking for natural language parsing. Tieu et al. [5] used boosting to choose a small number of features from millions of highly selective features for image retrieval. Blum et al. [6] proposed the co-training algorithm which trains a noise-tolerant learning algorithm using the noisy labels provided by another classifier.

In [7] we described a related co-retrieval approach which also attempted to learn the linear weights of different modalities with noisy label prediction. However, the current work represents several improvements over the previous algorithm: (1) The proposed algorithm is more efficient because it only trains on the top video clips provided by text retrieval instead of the whole collection; (2) It applies a unified boosting algorithm to select the most useful weak features with different loss functions; (3) An additional regularization step is added to avoid overfitting; (4) The positive and negative distributions are balanced before training; (5) It converts ranking features into further weak hypotheses.

4 Experiments

Our experiments followed the guidelines for the manual search task in the 2003 NIST TRECVID Video Track 2003[8], which require an automatic system to search without human feedback for video shots relevant to 25 query topics in a 65-hour news video collection. The retrieval units were video shots defined by a common shot boundary reference. The evaluation results are reported in terms

Fig. 4. The key frames of top 8 retrieved shots for query "Finding Tomb at Arlington National Cemetery". (a) Retrieval on text features (b) Co-Retrieval w/o image examples (c) Co-Retrieval with image examples

	Search w. Examples				Search w/o Examples			
Approaches	MAP	Prec10	Prec30	Prec100	MAP	Prec10	Prec30	Prec100
Text	0.157	0.292	0.225	0.137	0.157	0.292	0.225	0.137
Text/A/C	0.158	0.304	0.236	0.146	0.158	0.304	0.236	0.146
Global Oracle	0.188	0.368	0.259	0.16	0.164	0.336	0.235	0.152
CoRet+ExpLoss	0.206	0.444	0.307	0.171	0.177	0.352	0.261	0.156
CoRet+LogLoss	0.208	0.432	0.3	0.172	0.178	0.344	0.263	0.156
CoRet+RankLoss	0.207	0.448	0.301	0.172	0.178	0.344	0.26	0.156
CoRet+Truth	0.222	0.436	0.325	0.19	0.189	0.384	0.28	0.171
Local Oracle	0.285	0.512	0.344	0.199	0.212	0.436	0.304	0.171

Table 2. Comparison between various retrieval approaches. See text for details

of the mean average precision(MAP) and precision at top N retrieved shots. We generated 7 weak ranking features in our experiments including 4 types of general semantic features (face, anchor, commercial, outdoors), and 3 types of image-based features generated by the Euclidean distance of color, texture and edge histograms when query image examples were available. Detailed descriptions on the feature generation can be found in [7].

The following experiments consider two typical scenarios for video retrieval: 1. when only keywords are provided we use only semantic ranking features; 2. when both keywords and image examples are provided we additionally use image ranking features. The co-retrieval algorithm works as follows: first return at most 400 video shots using text retrieval as a base ranking function, label top $\alpha\%$ shots as positive and others as negative[2], learn the parameter λ based on the noisy labels and feed this back to the reranking model. We set the number of rounds T to be 10000 and choose the best round using cross validation. λ_0 is set to $n|\max_t \lambda_t|$, where n is number of weak hypotheses.

Figure 4 shows the performance improvement of Co-Retrieval without/with images examples over text retrieval alone. This improvement is achieved by successful reranking of top video shots. Table 2 lists a more detailed comparison for

[2] We augment noisy label prediction by reweighting shots identified as anchors or commercial from text retrieval scores. $\alpha\%$ is simply set to 25%, because our experiments show that retrieval performance is not very sensitive to the choice of $\alpha\%$.

	MAP	Prec10	Prec30	Prec100
ExpLoss	0.206	0.444	0.307	0.171
MAP	0.182	0.376	0.265	0.16

(a)

	MAP	Prec10	Prec30	Prec100
Reg	0.206	0.444	0.307	0.171
NoReg	0.192	0.404	0.293	0.171
More h	0.171	0.34	0.236	0.142

(b)

Table 3. Comparison between various retrieval approaches when image examples are available. (a) Co-Retrieval maximizing ExpLoss vs. MAP; (b) Co-Retrieval with regularization, without regularization and with automatically learned weak hypotheses

various retrieval approaches over mean average precision(MAP) at top 400 shots and precision at 10, 30 and 100 shots. Filtering out the anchor and commercial shots from text retrieval(**Text/A/C**) brings a slight performance improvement over text retrieval (**Text**). In contrast, Co-Retrieval with all three different loss functions (**CoRet+ExpLoss, LogLoss, RankLoss**) achieves a considerable and similar improvement over text retrieval in terms of all performance measures, especially when image examples are available MAP increases 5%. To investigate how noisy labels affect the results, we report the results of Co-Retrieval learning with truth labels(**CoRet+Truth**), which gives another 1.4% increase in MAP. This shows that the proposed algorithm is not greatly affected by the overfitting problem typical with noisy labels. We also report the results of two oracles using the algorithms presented in [1]: An oracle of the single best combination weight for all queries (**Global Oracle**) and an oracle for the optimal combination weights per query (**Local Oracle**), which assumes all relevant shots are known ahead of time. This analysis shows that the Co-Retrieval consistently performs better than the theoretical optimal fixed-weight combination.

5 Discussions

Why not optimize the performance criterion directly, that is mean average precision? Table 2 shows that there is a considerable performance gap between the local oracle and Co-Retrieval even with true labels. Therefore it is of interest to ask if we can optimize the performance criterion directly. However these performance criteria are usually not differentiable and not convex, which leads to several problems such as local maxima, inefficiency and poor generalization. Table 3(a) demonstrates the fact that maximizing mean average precision with noisy labels is not generalized enough to boost the true mean average precision.

Why is boosting not overfitting? It is well known that boosting type algorithms are not robust to noisy data and exhibit suboptimal generalization ability in the presence of noise, because it will concentrate more and more on the noisy data in each iteration[9]. However, our boosting algorithm does not seem to be affected by the overfitting problem even if our training data contains a lot of noise. Two answers come to mind. First, the regularization step improves generalizability which intentionally puts constraints on the choice of parameters. Table 3(b) compares the performance with and without regularization mentioned in Section 3.2 and shows that MAP will decrease about 1.4% without the regularization. Secondly, the version space of our weak hypotheses is much smaller than in most previous work such as [2], because we choose to fix the thresholds for

weak hypotheses instead of learning these thresholds automatically. Table 3(b) shows how performance is much worse when the threshold is allowed to learn. To explain this, we utilize a theoretical analysis of boosting algorithms by Schapire et al.[9]. They claim that a bound on the generalization error $P_{z\sim D}[\rho(z) \leq 0]$ depends on the VC-dimension d of the base hypothesis class and on the margin distribution of the training set. With probability at least $1-\delta$, it satisfies,

$$P_{z\sim D}[\rho(z) \leq 0] \leq P_{z\sim D}[\rho(z) \leq \theta] + \mathcal{O}\left(\frac{1}{\sqrt{l}}\left(\frac{d\log^2(l/d)}{\theta^2} + \log(1/\delta)\right)\right).$$

This analysis supports our observation that the generalization error will increase when the VC-dimension d becomes higher or equally the hypothesis space becomes larger. Learning flexible thresholds allows the algorithms to achieve lower empirical results for noise-free labels, however, in the highly noisy case, reducing the hypothesis space turns out to be a better choice for learning.

6 Conclusions

This paper presents a reranking framework for video retrieval to augment retrieval based on text features. We also propose a boosted reranking algorithm called Co-Retrieval, which applies the boosting type algorithm to automatically select the most useful weak hypotheses for different queries. Our experiments on the TRECVID 2003 search task demonstrates the effectiveness of the proposed algorithms whether or not image examples are available. Finally, we discuss two issues of Co-Retrieval on the choice of loss functions and the overfitting problem of boosting. As a possible extension, we can consider adding a relevance feedback function to the Co-Retrieval algorithm, which allows the interactive search system to rerank the current retrieval output given users' relevance feedback.

References

1. R. Yan and A. G. Hauptmann, "The combination limit of multimedia retrieval," in *Proc. of ACM Multimedia-03*, 2003.
2. Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," in *Proc. of ICML-98*, 1998, pp. 170–178.
3. M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," in *COLT*, 2000, pp. 158–169.
4. M. Collins, "Discriminative reranking for natural language parsing," in *Proc. 17th Intl. Conf. on Machine Learning*, 2000, pp. 175–182.
5. K. Tieu and P. Viola, "Boosting image retrieval," in *Intl. Conf. on Computer Vision*, 2001, pp. 228–235.
6. A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998.
7. A. G. Hauptmann et al, "Informedia at trecvid 2003: Analyzing and searching broadcast news video," in *Proc. of (VIDEO) TREC 2003*, Gaithersburg, MD, 2003.
8. TREC Video Track, "http://www-nlpir.nist.gov/projects/tv2003/tv2003.html," .
9. G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, Mar. 2001.