Video Search Reranking via Information Bottleneck Principle

Winston H. Hsu
Dept. of Electrical Engineering
Columbia University
New York, NY 10027, USA
winston@ee.columbia.edu

Lyndon S. Kennedy
Dept. of Electrical Engineering
Columbia University
New York, NY 10027, USA
lyndon@ee.columbia.edu

Shih-Fu Chang
Dept. of Electrical Engineering
Columbia University
New York, NY 10027, USA
sfchang@ee.columbia.edu

ABSTRACT

We propose a novel and generic video/image reranking algorithm, *IB reranking*, which reorders results from text-only searches by discovering the salient visual patterns of relevant and irrelevant shots from the approximate relevance provided by text results. The IB reranking method, based on a rigorous Information Bottleneck (IB) principle, finds the optimal clustering of images that preserves the maximal mutual information between the search relevance and the high-dimensional low-level visual features of the images in the text search results. Evaluating the approach on the TRECVID 2003-2005 data sets shows significant improvement upon the text search baseline, with relative increases in average performance of up to 23%. The method requires no image search examples from the user, but is competitive with other state-of-the-art example-based approaches. The method is also highly generic and performs comparably with sophisticated models which are highly tuned for specific classes of queries, such as named-persons. Our experimental analysis has also confirmed the proposed reranking method works well when there exist sufficient recurrent visual patterns in the search results, as often the case in multi-source news videos.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Performance, Experimentation

Keywords: Video Search, Multimodal Fusion, Information Bottleneck Principle

1. INTRODUCTION

Video and image retrieval has been an active and challenging research area thanks to the continuing growth of online video data, personal video recordings, digital photos, and 24-hour broadcast news. In order to successfully manage and use such enormous multimedia resources, users need to be able to conduct semantic searches over the multimodal corpora either by issuing text keyword queries or providing example video clips and images (or some combination of the two). Current successful semantic video search approaches usually build upon the text search against text associated with the video content, such as speech transcripts, close captions, and video OCR text. The additional use of other available modalities such as image content, audio, face detection, and high-level concept detection has been shown to improve upon the text-based video search systems [7, 20, 1, 17]. However, such multimodal systems tend to get the most improvement through leveraging multiple query example images, applying specific semantic concept detectors, or by developing highly-tuned retrieval models for specific types of queries, such as using face detection and speaker recognition for the retrieval of named persons. In the end, though, it will be quite difficult for the users of multimodal search systems to acquire example images for their queries. Retrieval by matching semantic concepts, though promising, strongly depends on availability of robust detectors and required training data. Likewise, it will be difficult for the developers of the system to develop highly-tuned models for every class of query and apply the system to new domains or data sets. It is clear, then, that we need to develop and explore approaches for leveraging the available multimodal cues in the search set without complicating things too much for the system users or developers.

Pseudo-relevance feedback (PRF) [20, 22, 15], is one such tool which has been shown to improve upon simple text search results in both text and video retrieval. PRF is initially introduced in [10], where the top-ranking documents are used to rerank the retrieved documents assuming that a significant fraction of top-ranked documents will be relevant. This is in contrast to relevance feedback where users explicitly provide feedback by labeling the top results as positive or negative. The same concept has been implemented in video retrieval. In [20], authors used the textual information in the top-ranking shots to obtain additional keywords to perform retrieval and rerank the baseline shot lists. The experiment was shown to improve MAP[1] from 0.120 to 0.124 (3.3% improvement) in the TRECVID 2004 video search task [21]. In [22], authors sampled the pseudo-negative images from the lowest rank of the initial query results; taking the query videos and images as the positive examples, the retrieval is then formulated as a classification problem which improves the search performance from MAP 0.105 to 0.112 (7.5% improvement) in TRECVID 2003. In

MM'06, October 23–27, 2006, Santa Barbara, California, USA.

[1]MAP: mean average precision, a search performance metric used in TRECVID. See more details in Section 6.1.

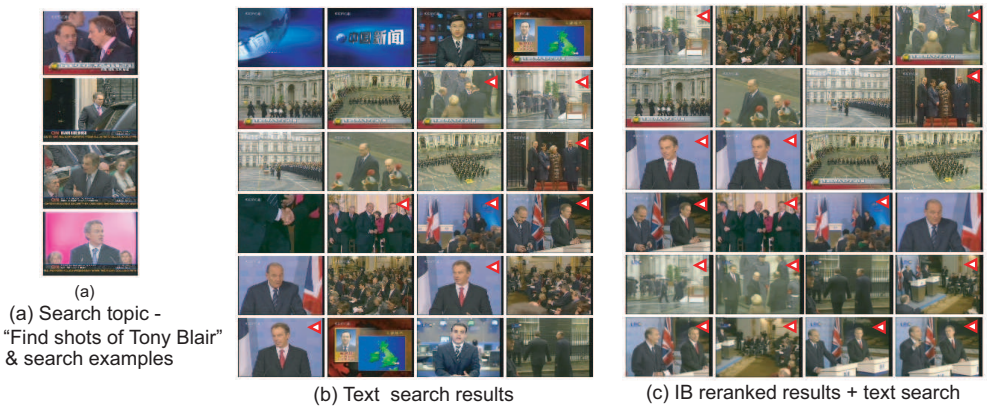

Figure 1: (a) TRECVID 2005 search topic 153, "Find shots of Tony Blair." (b) Top 24 returned shots from story-level text search (0.358 AP) with query terms "tony blair". (c) Top 24 shots of IB reranked results (0.472 AP) with low-level color and texture features. The red triangles mark the true positives.

[1], the authors made the assumption that very few images in the data set are actually relevant to the query and sampled pseudo-negative examples randomly from the data set. The pseudo-negative examples were used with the provided positive query examples to train multiple discriminative classifiers. Besides, authors of [5] used the Google image search return set as (pseudo-)positives and utilized a parts-based approach to learn the object model and then used that to rerank the search baseline images. The object model was selected, through a scoring function, among a large number ($\sim$100) of hypothesized parts-based models, which are very time consuming. Furthermore, their experiment was limited to image queries of simple objects such as bottles, cars, etc., instead of natural language queries as those in TRECVID.

The deficiency of current PRF approaches, however, is that the "visual" pseudo-positives are not utilized. Until now, only the textual information from the top-ranking or the visual pseudo-negatives from the lowest rank are considered. The reason is due to the poor accuracy of current video retrieval systems and the scarceness of true positive images in the top-ranking results (See examples in Figure 1). Because of this problem, existing systems avoid choosing the top-ranking video shots as pseudo-positives and rely on external visual search examples only [22]. However, the availability of video or image examples is another problem. For example, image search on Yahoo or Google allows textual queries only: users can not upload visual examples. In fact, many users would be reluctant or unable to provide such examples, anyway, as they can be difficult and time-consuming to find. When image examples are actually provided, they are usually quite sparse. The examples are few in number and fail to capture the relevant regions of high-dimensional feature space that might contain the true positive video shots. For example in Fig. 1, the query examples are not visually similar to many of the true positives and in some cases even some look like false positives.

In this work, to ease the problems of example-based approaches and avoid highly-tuned specific models, our goal is to utilize both the pseudo-positive and pseudo-negative examples and learn the recurrent relevant visual patterns from the estimated "soft" pseudo-labels. Instead of using "hard" pseudo-labels, the probabilistic relevance score of each shot is smoothed over the entire raw search results through kernel density estimation (KDE) [18]. An information-theoretic approach is then used to cluster visual patterns of similar semantics. We then reorder the clusters by the cluster relevance and then the images within the same cluster are ranked by feature density.

The semantic clustering approach is based on Information Bottleneck (IB) principle, shown to achieve significant performance gain in text clustering and categorization [14, 19]. The idea, as applied to text clustering, has been to use the information-theoretic optimization methodology to discover "cue word clusters," words of the same semantics, which can be used to represent each document at a mid level. The clusters are optimal in preserving the maximal mutual information (MI) between the clusters and the class labels.

Extended from the same information-theoretic property and based on our prior work [8], we propose a novel reranking method to find the best smoothed clusters which preserve the highest MI between (high-dimensional continuous) visual features and (hidden) search relevance labels. Meanwhile, we investigate multiple strategies to estimate the probabilistic relevance scores from initial search results. To balance reranking performance and efficiency, we experiment with different parameters used in IB ranking.

We tested the approach on the automatic video search tasks of TRECVID 2003-2005 and demonstrated its robust capability in boosting the baseline text search. Even without video/image search examples, the relative improvements, in terms of MAP, from the text baseline results are 20.8% in 2003, 17.7% in 2004, and 23.0% in 2005. In the TRECVID 2005 automatic search task, our text search plus IB reranking approach boosted the baseline story-level text retrieval to 0.107 MAP. This is a significant result, with comparable average performance to the state of the art using external example images (MAP 0.106, [1, 15]). By analyzing the experiment results over TRECVID 2005 data, we observed that the proposed IB reranking methods worked best for named people query topics. This is intuitive, as there are usually recurrent videos of named subjects in the news across multiple broadcast channels. The IB reranking method takes advantage of such patterns in improving the rankings. With the same rationale, the method expectably suffers performance loss for a small number of topics when such repeated patters lack. But in general the average performance over all search

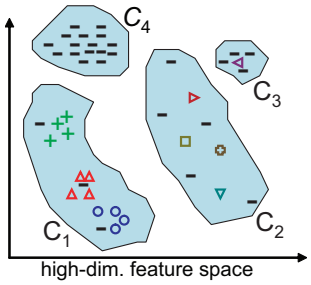

Figure 2: An example of 4 search relevance-consistent clusters C_1 to C_4, where C_1 has the highest denoised posterior probability $p(y = 1|c)$ and C_4 has the lowest. "–" is a pseudo-negative and the others are pseudo-positives. Pseudo-positives in the same shape are assumed having similar appearances. Note that the "hard" pseudo-labels are for illustration only; instead we use "soft" labels in this work.

topics has shown significant improvement. Furthermore, by applying the class dependent query method [11], we may apply the proposed reranking method adaptively to the query class (e.g., named people) that are predicted to benefit most from reranking.

Moreover, IB reranking is highly generic and requires no training in advance but is comparable to the top automatic or manual video search systems many of which are highly tuned to handle named-person queries with sophisticated models such as face detection and speaker identification (cf. Section 6.5).

We provide the rationale for the approach in Section 2. The main idea of the IB principle and its extension to high-dimensional continuous random variables are introduced in Section 3. The IB reranking algorithm is described in Section 4. We describe the feature representations and text search in Section 5. Evaluation of the proposed techniques on TRECVID video search tasks is detailed in Section 6. We present conclusions and future work in Section 7.

2. MOTIVATION: VIDEO RERANKING

In video search systems, we are given a query or a statement of information need and we then need to estimate the relevance $R(x)$ of each video shots in the search set, $x \in X$, and order them by their relevance score. Many approaches have been tested in recent years, ranging from plainly associating video shots with text search scores to sophisticated fusion of multiple modalities. Some approaches rely on user-provided query images as positive examples to train a supervised classifier to approximate the posterior probability $P(Y|X)$, where Y is a random variable representing search relevance [15]. The posterior probability is then used for $R(x)$ in video ranking.

There are certain problems and limitations for example-based video search approaches as discussed in Section 1. To overcome these problems, we observe that text search results can generally bring up certain relevant videos near the top of the return set (i.e., Fig. 1-(b)). In a large image or video database, in particular, some positive images may share great visual similarity, but receive quite different initial text search scores. For example, Fig. 1-(b) are the top 24 shots from story-level text search for TRECVID 2005 topic 153, "Find shots of Tony Blair". We can see recurrent relevant shots, of various appearances, dispersed among the return set but mixed with some irrelevant ones (e.g., anchor or graphic shots).

Such recurrent images or videos are commonly observed in image search engines (e.g., Yahoo or Google) and photo sharing sites (e.g., Flickr). Interestingly, authors of [9] had quantitatively analyzed the frequency of such recurrent patterns (in terms of visual duplicates) for cross-language topic tracking – a large percentage of international news videos share re-used video clips or near duplicates. Such visual patterns were used in [9] for tracking topics and will be the basis for search result reranking in this paper.

According to the above observation, instead of using user-provided search examples, we argue to consider the search posterior probability, estimated from initial text-based results in an unsupervised fashion, and the recurrent patterns (or feature density) among the image features and use them to rerank the search results. A straightforward implementation of the idea is to fuse both measures, search posterior probability and feature density, in a linear fashion. This fusion approach is commonly used in multimodal video search systems [4]. It can be formulated as the following:

$$R(x) = \alpha p(y|x) + \beta p(x), \tag{1}$$

where $p(y|x)$ is the search posterior probability and $p(x)$ is the feature density of the retrieved images[2] and α and β are scalars for linear fusion.

The above equation incurs two main problems, which are confirmed in our experiments (cf. Section 6.2 or Table 1). The posterior probability $p(y|x)$, estimated from the text query results and (soft) pseudo-labeling strategies, is noisy; a "denoised" representation for the posterior probability is required. Besides, the feature density estimation $p(x)$ in Eqn. 1 may be problematic since there are usually frequent recurrent patterns that are irrelevant to the search (e.g., anchors, commercials, crowd scenes, etc.). Instead, we should consider only those recurrent patterns within buckets (or clusters) of higher relevance.

To exploit both search relevance and recurrent patterns, we propose to represent the search relevance score $R(x)$ as the following:

$$R(x) = \alpha p(y|c) + \beta p(x|c), \tag{2}$$

where $p(y|c)$ is a "denoised" posterior probability smoothed over a relevance-consistent cluster c, which covers image x, and $p(x|c)$ is the local feature density estimated at feature x. The cluster denoising process has been shown effective in text search [13]. Meanwhile, the local feature density $p(x|c)$ is used to favor images that occur multiple times with high visual similarity. Choices of parameters α and β will affect the reranked results. In the preliminary experiments, we observed that the denoised posterior probability $p(y|c)$ is more effective and plays the main role for search relevance when compared to the pattern recurrence within the same relevant clusters. Accordingly, an intuitive approach is to let α be significantly larger than β so that the reranking process first orders clusters at a coarse level and then refines the order of images in each cluster according to local feature density. The effectiveness of such an approach will be verified in the experiment section.

[2]In this work, visual features are extract from key-frames of each video shot.

Two main issues arise in the above proposed approach: (1) how to find the relevance-consistent clusters, in an unsupervised fashion, from noisy text search results and high-dimensional visual features; (2) how to utilize the recurrent patterns across video sources. To address the first problem, we adopt the IB principle, which finds the optimal clustering of the images that preserves the maximal mutual information about the search relevance. The denoised posterior probabilities $p(y|c)$ are iteratively updated during the clustering process. The feature densities $p(x|c)$ are then estimated from each cluster c accordingly.

The idea is exemplified in Fig. 2, where 4 relevance-consistent clusters are discovered automatically. Images of the same cluster (i.e., C_1) have the same denoised posterior probability $p(y|c)$, but might have recurrent patterns of different appearances. For example, C_1 has 3 different regions which have high density in the feature space. We first rank the image clusters by $p(y|c)$ and then order within-cluster images by the local feature density $p(x|c)$. In short, those visually consistent images which occur the most frequently within higher-relevance clusters will be given higher ranks.

3. THE IB PRINCIPLE

The essence of the IB reranking approach (cf. Section 4.1) is to smooth the noisy text search results in the high-dimensional visual feature space and to find relevance-consistent clusters, which was first developed in our prior work [8]. We will give an overview in this section and further extend them to video search problems.

The variable X represents features and Y is the variable of interest or auxiliary labels associated with X. X might be documents or low-level feature vectors; Y might be document types in document categorization or sematic class labels, or search relevance. In this context, we want the mapping from $x \in X$ to cluster $c \in C$ to preserve as much information about Y as possible. As in the compression model, the framework passes the information that X provides about Y through a "bottleneck" formed by the compact summaries in C. On the other hand, C is to catch the consistent semantics of X. The semantic is defined by the conditional distribution over the label Y (i.e., $p(y|x)$).

The above goal can be formulated by the IB principle, which states that among all the possible clusterings of the objects into a fixed number of clusters, the optimal clustering is the one that minimizes the loss of mutual information (MI) between the features X and the auxiliary labels Y. Assume that we have joint probability $p(x, y)$ between these two random variables. According to the IB principle, we seek a clustering representation C such that, given a constraint on the clustering quality $I(X;C)$, the information loss $I(X,Y) - I(C;Y)$ is minimized.

3.1 Mutual Information

For discrete-valued random variables X and Y, the MI [3] between them is

$$I(X;Y) = \sum_y \sum_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)}.$$

We usually use MI to measure the dependence between variables. In the IB framework, we represent the continuous D-dimensional features with random variable $X \in R^D$; the auxiliary label is a discrete-valued random variable Y representing the target or relevance labels. We have feature observations with corresponding labels in the training set $S = \{x_i, y_i\}_{i=1..N}$. Since X is continuous, the MI is

$$I(X;Y) = \sum_y \int_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dx.$$

However, based on S, the practical estimation of MI from the previous equation is difficult. To address this problem, the histogram approach is frequently used but only works for scalar-valued variables. An alternative approach is to model X through Gaussian Mixture Model (GMM) which is limited to low-dimensional features due to the sparsity of data in high-dimensional spaces [6].

We approximate the continuous MI with Eq. 3 for efficiency. The summarization is only over the observed data x_i assuming that $p(x, y) = 0$ if $x \notin S$. Similar assumptions are used in other work (e.g., the approximation of Kullback-Leibler divergence in [6]). According to our experiments, the approximation is satisfactory in measuring the MI between the continuous feature variable X and the discrete auxiliary variable Y.

$$I(X;Y) \cong \sum_{x_i \in S} \sum_y p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} \quad (3)$$

3.2 Kernel Density Estimation

To approximate the joint probability $p(x, y)$ based on the limited observations S, we adopt the kernel density estimation [18]. The method does not impose any assumption on the data and is a good method to provide statistical modeling among sparse or high-dimensional data.

The joint probability $p(x, y)$ between the feature space X and the auxiliary label Y is calculated as follows:

$$p(x,y) = \frac{1}{Z(x,y)} \sum_{x_i \in S} K_\sigma(x - x_i) \cdot \bar{p}(y|x_i), \quad (4)$$

where $Z(x, y)$ is a normalization factor to ensure $\sum_{x,y} p(x, y) = 1$, K_σ (Eq. 5) is the kernel function over the continuous random variable X. $\bar{p}(y|x_i)$ is an un-smoothed conditional probability of the auxiliary labels as observing feature vector x_i. We assume that Y is binary in this experiment and $\bar{p}(y|x_i)$ can be assigned by considering different strategies discussed in Section 4.2. Note that Y can extend to multinomial cases in other applications.

From our observation, $\bar{p}(y|x_i)$ is usually sparse. Eq. 4 approximates the joint probability $p(x, y)$ by taking into account the labels of the observed features but weighted and smoothed with the Gaussian kernel, which measures the non-linear kernel distance from the feature x to each observation x_i. Intuitively, nearby features in the kernel space will contribute more to Eq. 4.

Gaussian kernel K_σ for D-dimensional features is defined as:

$$K_\sigma(x_r - x_i) = \prod_{j=1}^{D} \exp \frac{-||x_r^{(j)} - x_i^{(j)}||}{\sigma_j}, \quad (5)$$

where $\sigma = [\sigma_1, .., \sigma_j, .., \sigma_D]$ is the bandwidth for kernel density estimation. We can control the bandwidth to embed prior knowledge about the adopted features; for example, we might emphasize more on color features and less on the texture features by changing the corresponding σ_j.

3.3 Sequential IB Clustering

We adopt the sequential IB (sIB) [14] clustering algorithm to find optimal clusters under the IB principle. It is observed that sIB converge faster and is less sensitive to local optima compared to other IB clustering approaches [14].

The algorithm starts from an initial partition C of the objects in X. The cluster cardinality $|C|$ and the joint probability $p(x, y)$ are required in advance. We will discuss the selection of $|C|$ in Section 6.4. At each step of the algorithm, one object $x \in X$ is drawn out of its current cluster $c(x)$ into a new singleton cluster. Using a greedy merging criterion, x is assigned or merged into c^* so that $c^* = \text{argmin}_c\, d_F(\{x\}, c)$. The merging cost, the information loss due to merging of the two clusters, represented as $d_F(c_i, c_j)$, is defined as (cf. [19] for more details):

$$d_F(c_i, c_j) = (p(c_i) + p(c_j)) \cdot D_{JS}[p(y|c_i), p(y|c_j)], \qquad (6)$$

where D_{JS} is Jensen-Shannon (JS) divergence and $p(c_i)$ and $p(c_j)$ are cluster prior probabilities. JS divergence is non-negative and equals zero if and only if both its arguments are the same and usually relates to the likelihood measure that two samples, independently drawn from two unknown distributions, are actually from the same distribution.

The sIB algorithm stops as ϵ, the ratio of new assignments among all objects X to new clusters, is less than a threshold, which means that the clustering results are "stable" and no further reassignments are needed. Decreasing the threshold will cause the algorithm to take more time and more iterations to find "stable" clusters and increase $I(C; Y)$. The impact of ϵ on time complexity and reranking performance is discussed in Section 4.3.

Multiple random initializations are used to run sIB multiple times and select the result that has the highest cluster MI $I(C; Y)$, namely the least information loss $I(X; Y) - I(C; Y)$.

As in any clustering problem, determining the number of clusters is non-trivial. In [8], we utilized the MI distortion to locate the most significant cluster number for given items. The approach is rigorous but requires extensive computation. Instead, for this experiment, we set the cluster number $K = \lceil \frac{N}{N_c} \rceil$, where $N_c = 25$ is a preset expected number of items in each cluster and is empirically determined through validation experiments. More details about determining the cluster number are in Section 6.4.

3.4 Cluster Conditional Probability

During each cluster merging or splitting process, as described in Section 3.3, for each cluster c, we should update its cluster conditional probability $p(y|c)$, which is also an input of JS divergence of Eqn. 6.

$$p(y|c) = \frac{1}{p(c)} \sum_{x \in c} p(x, y), \qquad (7)$$

where $p(x) = \sum_y p(x, y)$ and $p(c) = \sum_{x \in c} p(x)$ is the cluster prior. See more explanations in [19].

4. IB RERANKING APPROACH

4.1 Reranking Steps

IB reranking reorders the search results derived from other search mechanisms (e.g., text search). The inputs are lists of shot or image indices J_i and their corresponding text search

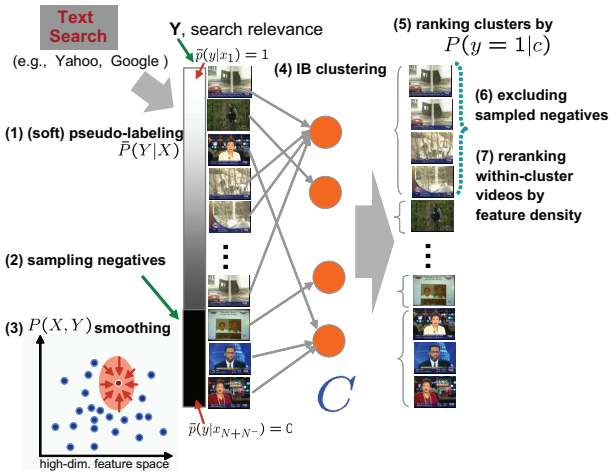

Figure 3: The IB reranking steps from baseline text search. See details in Section 4.1.

relevance scores s_i (if available) and can be represented as $\{(J_i, s_i)\}$. Note that the output score s_i might be unavailable in some cases such as the image return sets from Google or Yahoo searches. For IB reranking, the hidden variable Y is the search relevance random variable we are interested in. The features $x_i \in X$ for each item J_i can be derived accordingly. The major steps of IB reranking are as follows (also illustrated in Fig. 3):

1. Estimate (soft) pseudo-labels and treat them as un-smoothed conditional probability $\bar{p}(y|x_i)$ (cf. Section 4.2) from the top N images $\{(J_i, s_i)\}$ of text search output.
2. Sample N^- negative examples from the database and set $\bar{p}(y = 1|J) = 0$ for these negative examples.
3. Calculate smoothed joint probability $p(x, y)$ through Eqn. 4 for these $N + N^-$ items.
4. Conduct sIB clustering (cf. Section 3.3) on $N + N^-$ images into K clusters with convergence threshold ϵ.
5. Rerank clusters by search relevance, cluster conditional probability $p(y|c)$ (See Eqn. 7), in descending order.
6. Exclude those N^- sampled negative examples.
7. Rerank images within each cluster c in descending order by feature density estimated as follows[3]:

$$p(x_j|c) = \frac{1}{Z_c} \sum_{x_k \in c, k \neq j} K_\sigma(x_k - x_j). \qquad (8)$$

8. Output the ordered list $J_{i'}$.

The aim of the IB reranking approach is to statistically learn the recurrent relevant patterns and reorder them according to the denoised cluster conditional probability $p(y|c)$. We first estimate the un-smoothed conditional probability $\bar{p}(y|x_i)$ from the initial text search output. Then we smooth it through the whole feature space approximated by the top

[3] Z_c in Eqn. 8 is a normalization factor to ensure $\sum_j p(x_j|c) = 1$.

N images and N^- sampled negatives. This is reasonable since the pseudo-labeling approach is just approximating the correlation between the semantic (search relevance, Y) and feature space X. Intuitively, those low-ranked images or sampled negative examples are helpful [22, 15] to push down the false positives in the top of text output. Vice versa, salient top-ranked positive patterns can pull up other relevant but low-ranked images from the preliminary text search. Such examples are seen in Fig. 1-(b) and (c). Even though the anchor/graphics images appear frequently in the top of the text results, they also appear equally as frequently throughout the bottom of the results or sampled negative images. The smoothing process tends to push them down the search list.

The sIB clustering further groups images of similar relevance into consistent clusters and computes its corresponding denoised cluster conditional probability $p(y|c)$, which is another smoothing process by using the cluster posterior $p(y|c)$ to replace the individual posterior $p(y|x)$.

The within-cluster images are further ordered by their local feature density $p(x|c)$ estimated from the items of the same cluster, where we assume that images with more frequently recurrent patterns are more relevant than isolated dissimilar images in the same cluster. Note that in all the reranking steps we need not have "hard" positive or negative samples, which are required in prior example-based approaches, but only their soft un-smoothed conditional probability $\bar{p}(y|x)$ available from initial text-based search.

4.2 Pseudo-labeling Strategies

The IB reranking method estimates the un-smoothed search relevance probability, $\bar{p}(y|x_i)$, by using the initial text search score s_i. x_i is the image feature of image J_i and $y \in Y$ is the relevance random variable. We experimented with three different strategies for such estimation of "soft" pseudo-labeling.

4.2.1 Binary

In the "binary" approach, we estimate the un-smoothed search relevance probability of image J_i with text search score s_i in a binary form.

$$\bar{p}(y = 1|x_i) = 1_{\{s_i \geq e_s\}},$$

where $1_{\{\cdot\}}$ is an indication function and e_s is the search score threshold. Empirically, one can use the mean plus one standard deviation of the entire text search scores. Or one could use cross-validation experiments to set a suitable e_s value.

4.2.2 Normalized Rank

For certain cases when text search scores are unavailable and only ranking orders are given, we adopt the normalized rank [22] to estimate $\bar{p}(y|x_i)$ of image J_i, which is the i'th ranked image:

$$\bar{p}(y = 1|x_i) = 1 - \frac{i}{N},$$

where N is the number of return set to be reranked from the initial text search output. Naturally, the first-ranked image will be $\bar{p}(y|x) = 1$ and the last will be 0.

4.2.3 Score Stretching

In the "score stretching" approach, $\bar{p}(y = 1|x_i)$ is estimated by setting the middle point (0.5) at the text search score threshold e_s and linearly stretching those scores above or under e_s to be within $[0, 1]$.

$$\bar{p}(y = 1|x_i) = \frac{1}{2} + 1_{\{s_i \geq e_s\}} \cdot \frac{s_i - e_s}{2(max_s - e_s)} - 1_{\{s_i < e_s\}} \cdot \frac{e_s - s_i}{2(e_s - min_s)},$$

where max_s and min_s is the maximum and minimum text search scores.

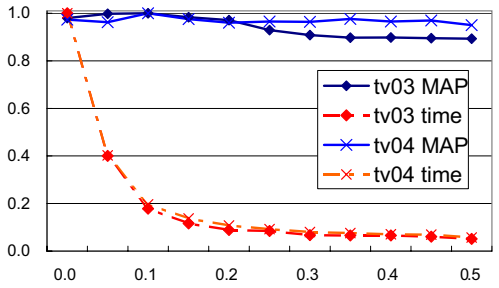

Figure 4: Time complexity vs. reranking performance, in a normalized scale, on TRECVID 2003 (tv03) and 2004 (tv04) data sets. Note that "score stretching" pseudo-labeling approach is used. See explanations in Section 4.3

4.3 Reranking Complexity vs. Thresholds

The sIB convergence threshold ϵ affects the time complexity for sIB clustering (cf. Section 3.3) and the clustering quality, in terms of MI distortion, and ultimately will influence the reranking performance. A lower threshold ϵ will force the algorithm to take a longer time to converge. This usually means a more "stable" clustering result. However, we are curious about the trade-off of time complexity and reranked performance. We tested the reranking process at different thresholds ranging from 0.01 to 0.5 on both TRECVID 2003 and 2004 queries. Results of time complexity vs. reranking performance, in a normalized scale, are shown in Fig. 4. The experiments reranked 1250 items, 1000 (N) video shots plus 250 (N^-) sampled negative examples. Increasing the threshold sharply reduces the clustering time but only degrades the performance slightly in both TRECVID data sets. A setting of $\epsilon = 0.20$ reduces the computation time by 10 folds while keeping almost unchanged performance (MAP). It suggests that most of the relevant or irrelevant data are in the right order after just a few sIB iterations. In the following experiments, we fix the threshold $\epsilon = 0.20$. Implemented in MATLAB on a regular Intel Pentium server, it takes around 18 seconds to rerank a query of 1250 images.

5. FEATURE REPRESENTATIONS

5.1 Low-level Features

For low-level features X, we represent each key-frame with a 273 dimensional feature vector composed of two low-level global visual features. The first is color moments over 5x5 fixed grid partitions [1], where the first 3 moments of 3 channels of CIE Luv color space are extracted; it results in a 225-dimensional feature vector per key-frame. The second is Gabor texture [1] where we take 4 scales and 6 orientations of Gabor transformation and further use their means and standard deviations to represent the whole key-frame

and result in a 48-dimensional feature. Though they are basic image features, prior work such as [17, 1] has shown their excellent performance in image retrieval and semantic concept detection. To analyze the contribution of each feature, we have also evaluated the reranking performance by using grid color moments only, which leads to a relative performance drop of 8% compared to using both color and texture features described above.

5.2 Text Search

IB reranking is built on top of text search against the text (e.g., ASR or machine translation) transcripts of the videos in the database. We specially choose the best possible text search approach to provide the best baseline for improvement with IB. The text searches are conducted automatically using the natural language statements of information need, provided by NIST [21]. The natural language queries (such as "Find shots of Iyad Allawi, the former prime minister of Iraq" or "Find shots of a ship or boat") are parsed through part-of-speech tagging [12] and named entity tagging [2]. Keywords (like "ayad allawi iraq" and "ship boat") are extracted. If named entities exist, then those are chosen as the keywords. If not, then the nouns are chosen. Finally, if no nouns or named entities are present, then the main verb is chosen. A standard list of stop words is also removed. The extracted keywords are then used to issue queries against the speech recognition transcripts using the Okapi BM-25 formula [16] and all terms are stemmed using Porter's algorithm.

Since the retrieval unit in video search is the shot (a single continuous camera take), there arises the problem of how to associate chunks of text with each video shot. A simple approach might be to take the text document for each shot to be the text that is contained temporally within the boundaries of the shot. Or, we could compensate for the asynchrony between the speech transcripts and the video stream by including some buffer text from a fixed window before and after the shot as well. Our experiments, however, have shown that the best approach is to use the text from the entire story within which the shot is contained. This makes sense since the true semantic relationships between images and the text transcript exist at the story level: if a concept is mentioned in the text it is likely to appear in the video stream somewhere within the same story, but it is unlikely to appear in the next story or the previous one. Story boundaries can be extracted (imperfectly, but with reasonable reliability) automatically through the visual characteristics of the video stream and the speech behavior of the anchorperson. Different story boundary detectors are trained separately for each language – English, Chinese, and Arabic. The performance, evaluated with TRECVID metrics (F1[4]), is 0.52 in English, 0.87 in Arabic, and 0.84 in Chinese [8]. Our experiments have shown that choosing text within automatically detected story boundaries to associate with shot documents outperforms the fixed-window based approach consistently with approximately 15% improvement in terms of MAP across the TRECVID 2003, 2004, and 2005 data sets. Using manually annotated story boundaries offers an additional 5-10% increase in performance. Note that automatically detected boundaries are used in this work.

[4] $F1 = \frac{2 \cdot P \cdot R}{P+R}$, where P and R are precision and recall rates defined in TRECVID [21] story boundary detection task.

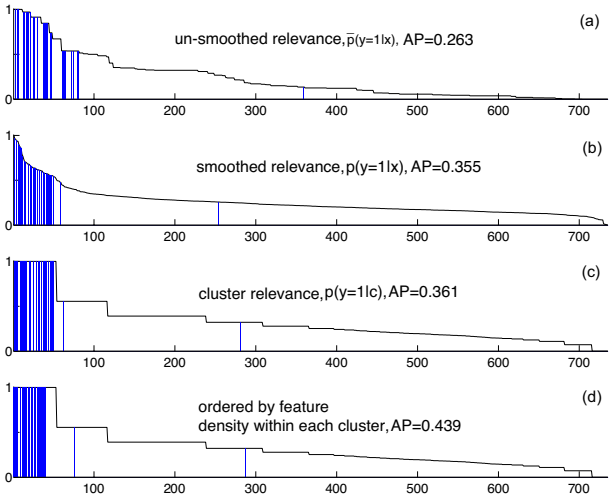

Figure 5: Reranking steps and their corresponding AP of topic 171, "Find shots of a goal being made in a soccer match"; (a)-(c) are normalized scores of video shots, ordered by their corresponding measures $\bar{p}(y = 1|x)$, $p(y = 1|x)$, and $p(y = 1|c)$; In (d), shots in the cluster (with the same $p(y = 1|c)$) are ordered by feature density. The blue lines mark the true positives.

The use of entire stories for retrieval gives an increase in recall (more of the true relevant shots are found), but gives a decrease in precision (more noise also turns up). This provides an excellent motivation for the application of IB reranking, since, if text search is working well, then many of the relevant documents are found and ranked highly and we can exploit a method to mine the salient visual patterns from those shots and push down the noisy irrelevant shots out of the top-ranked spots.

6. EXPERIMENTS

6.1 Data Set

We conduct the experiments on TRECVID 2003-2005 data sets [21]. In TRECVID 2003-2004, there are 133 and 177 hours of videos respectively with English ASR transcripts both from CNN and ABC channels in 1998. The TRECVID 2005 data set contains 277 international broadcast news videos which include 171 hours of videos from 6 channels in 3 languages (Arabic, English, and Chinese). The time span is from October 30 to December 1, 2004. The ASR and machine translation (MT) transcripts are provided by NIST [21].

For the performance metric we adopted non-interpolated average precision (AP), which corresponds to the area under an (non-interpolated) recall/precision curve. Since AP only shows the performance of a single query, we use mean average precision (MAP), which is the mean of APs for multiple queries, to measure average performance over sets of different queries in a test set. See more explanations in [21].

6.2 Breakdowns in Reranking Steps

Before we present the average performance over all queries, let's analyze in depth an example query to understand the

contributions from each step of the IB reranking process. For search topic 171, "Find shots of a goal being made into a soccer match," terms "goal soccer match" are automatically determined and used to derive the text search results. Fig. 5-(a) are scores of video shots ordered by the estimated un-smoothed conditional probability $\bar{p}(y|x)$ through "score stretching" pseudo-labeling strategy (cf. Section 4.2.3). The AP is the same as the original text search score. Fig. 5-(b) are those by the smoothed conditional probability $p(y|x) = \frac{p(x,y)}{p(x)}$, estimated from the top N^+ text return set and N^- sampled pseudo-negatives. The smoothing process can bring up some positives or push down negatives, and hence improve AP accordingly. This can be confirmed by the fact that most true positives (as blue lines in Fig. 5-(b)) are moved to higher ranks. Interestingly, through the sIB clustering approach, almost all of the recurrent relevant scenes, though of diverse appearances, are clustered in the same and the most relevant cluster, as shown in Fig. 5-(c), where the plateau represents the same $p(y|c)$ within the same cluster. Further ordered by feature density $p(x|c)$, the relevant shots are further pushed to the top, as shown in Fig. 5-(d).

As shown in Fig. 5, exceptions can also be found. Not all true positives are included in the first cluster (See 2 sparse blue lines in Fig. 5-(c) and (d)). Such cases may actually become worse after reranking. However, as supported by the overall performance comparisons (Table 1), the majority of results for most queries benefit from the reranking process. It is also important to note that the quality of top results are most important for search tasks, as intuitively captured by the definitions of AP performance metric.

The contributions from the major reranking steps averaging across all official queries of TRECVID 2005 are listed in Table 1. Consistent improvements by IB reranking are confirmed[5]. Besides, relevance probability smoothing has a large impact on performance (13.5% gain over initial text search). Fusion with prior (as in Eqn. 1) without clustering actually hurts (performance gain dropped to 8.3%). The proposed IB reranking achieves a significant gain at 20.1%, which is increased to 22.4% if IB reranking results are further fused with initial text search results.

6.3 Performance on All TRECVID Queries

We conduct IB reranking on all queries of TRECVID 2003-2005. We first compared the three pseudo-labeling strategies on both TRECVID 2003 and 2004 data sets. As shown in Table 2, "score stretching" approach is the most effective and is later solely used in the TRECVID 2005 test, since it naturally utilizes the continuous relevance output from the text modality; IB reranking improves the performance (MAP) of text search baseline and up to 23%.

The performance (story text vs. story text + IB reranking) in AP across all queries is listed in Fig. 6, where IB reranking improves or retain the same performance of text search baseline for a large majority of queries. IB reranking benefits the most for queries with salient recurrent patterns; i.e., "Sam Donaldson" (135), "Omar Karami" (151), "Blair"

[5]Note that the scores in Table 1 and Fig. 5 are slightly lower than those in Table 2 since less than 1000 shots are used for the evaluation. The reranking process are applied on 1000 sub-shots which need to be merged into shots. It is a requirement for TRECVID [21]. The number of shots are less than 1000 after the merging.

#	steps/measures, $R(x)$	MAP	improvement
1	$\bar{p}(y\|x)$	.0858	0.0%
2	$p(y\|x)$	.0974	13.5%
3	$\alpha p(y\|x) + \beta p(x)$	.0930	8.3%
4	IB reranking	.1031	20.1%
5	IB reranking+text	.1050	22.4%

Table 1: The performance breakdowns in major reranking steps evaluated in TRECVID 2005 queries. The absolute MAPs and relative improvements from the text baseline are both shown. "IB reranking+text" means that the reranking results are fused with the original text scores via ranking order fusion method. Row 3 lists the best MAP among sets of (α, β) in the implementation of Eqn. 1. $\bar{p}(y|x)$ is the initial search relevance from text search scores. $p(y|x)$ is a smoothed version of $\bar{p}(y|x)$. See more explanations in Section 6.2.

(153), "Mahmoud Abbas" (154), "baseball" (102), "Spinx" (116), "Down Jones" (120), and "soccer" (171). This makes sense since the approach, though requiring no example images, tries to infer the recurrent patterns highly relevant to the search relevance based on the initial search scores and visual density estimation. Specifically, the visual patterns present in the search results will help boost the posterior probabilities of relevant data through denoising (Eqn. 7) and local density based reranking in each cluster (Eqn. 8).

Fig. 6 also shows several query topics where performance is degraded after IB reranking. The queries include "building-fire" (147), "Pope" (123), and "Boris Yelstin" (134). Upon further examination, we found the relevant videos for such queries are either of a small number or lack consistent visual patterns. For example, scenes of the Pope are actually of different events and thus do not form consistent visual appearances. This explains why IB reranking does not provide benefits from such queries.

IB reranking requires no external image examples but just reranks images from the text output directly. This is an important advancement in video search since users do not have or are reluctant to provide image examples. Surprisingly, the novel approach is competitive with and actually complementary to those state-of-the-art example-based search approaches (cf. Section 6.6 and 6.5). Nevertheless, visual examples significantly outperform text search in certain queries such as "tennis" (156), "basketball" (165), etc. This offers a promising direction for further expanding the proposed approach – as external example images are available, we may consider embedding these "true" positives in the IB framework for reranking; i.e., setting the un-smoothed conditional probability $\bar{p}(y|x) = 1$ for example images.

6.4 Number of Clusters

The number of clusters is an important parameter for clustering algorithms. Our prior work [8] used an information-theoretic measure to determine the optimal cluster number of the "visual cues," which are later used as bases for a new feature representation. To avoid the large computational cost, we empirically select the best cluster number threshold N_c through multiple experiment runs. We experimented with different cluster thresholds over TRECVID 2003 and 2004 data sets and choose the one that resulted in the best

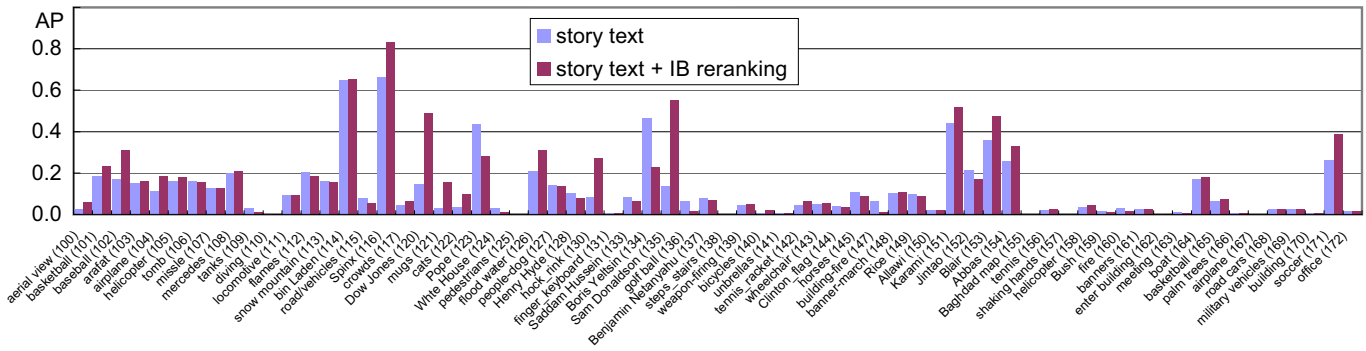

Figure 6: Performance of IB reranking and the baseline text search across all queries of TRECVID 2003-2005.

Exps./Stratgies	text baseline	binary	normalized rank	score stretching
TRECVID 2003	0.169	0.187 (10.6%)	0.177 (5.0%)	0.204 (20.8%)
TRECVID 2004	0.087	0.089 (1.7%)	0.098 (12.9%)	0.102 (17.7%)
TRECVID 2005	0.087	–	–	0.107 (23.0%)

Table 2: IB reranking performance (top 1000 MAP) and comparison with the baseline (story) text search results of TRECVID 2003, 2004, and 2005. Each column uses a different pseudo-labeling strategy. Percentages shown in parentheses are improvement over the text baseline.

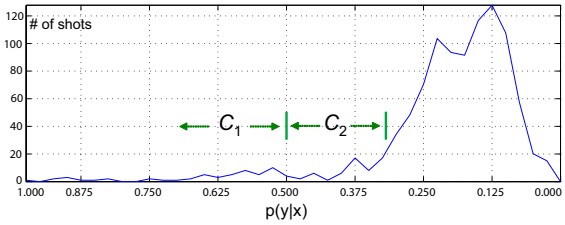

Figure 7: Histogram of normalized search posterior probability $p(y = 1|x)$ from top 1000 shots of topic 171. The top clusters (C_1 and C_2) correspond to natural cut points in terms of (estimated) search relevance scores.

performance. Then for the new data set in TRECVID 2005, we applied the same cluster threshold without readjustment to assure the generality of the empirical choice. We have found a cluster number corresponding to an average cluster size $N_c = 25$ would provide satisfactory results.

In addition, we have found the reranking performance is not sensitive to the choice of the cluster numbers. In most queries, the most critical results (in terms of AP calculation and user satisfaction) are in the first few clusters or pages. Thus, slight changes of the number of clusters will not significantly affect the distribution of such top results.

In Fig. 7., we show a histogram, the number of video shots, over normalized search posterior probability $p(y = 1|x)$ (cf., Fig. 5-(b)) from top 1000 images of topic 171. The first two most relevant clusters C_1 and C_2 are also labeled. The results indicate that the IB clusters are effective and intuitive – the top clusters coincide with the intuitive cutoff points (0.5 and kneeling point).

6.5 Performance on Named-Person Queries

IB reranking works best when initial text search results contain a reasonable number of positive shots and the positive shots are somewhat visually related. Such conditions often occur when the video sources come from multiple concurrent channels reporting related events. This is more the case for TRECVID 2005 than TRECVID 2003 and 2004.

In the TRECVID 2005 benchmark data, we have observed that IB reranking works particularly well on queries for named persons. This is surprising since the visual content of example images and shots has not previously been shown to be very helpful for retrieval of named persons. For example, fusing simple story-based text search with a high-performing content-based image retrieval (CBIR) system [1, 15] provides only modest gains over story-based text search on the six named person queries in the TRECVID 2005 set. Story-based text search results in a MAP of 0.231 over these six queries, while fusing with the CBIR system [1, 15] provides a very minor improvement in MAP to 0.241, a relative improvement of 4%. On the other hand, if we apply IB reranking after story-based text search, we get an improvement in MAP to 0.285, a big improvement of over 23%. So, IB reranking is able to capture the salient visual aspects of news events contained in the search set in which particular named people appear, which is very difficult to do with example images which come from sources other than the search set or from a different time span. When compared to the performance of all official automatic and manual submissions on the six named person queries, illustrated in Fig. 8, IB reranking outperforms all manual runs and is second (but comparable) to only one automatic run (MAP: .299), which is highly tuned to handle named person queries with face detection and other models requiring external resources. However, the IB approach is highly generic and requires no training (specific to the named person search) in advance.

6.6 Class-Dependent Fusions

We have seen that the benefit of IB reranking is significant on named person queries. Similar to prior work in class-dependent query [11], we also tested a query-class-dependent model, where different approaches can be used for different

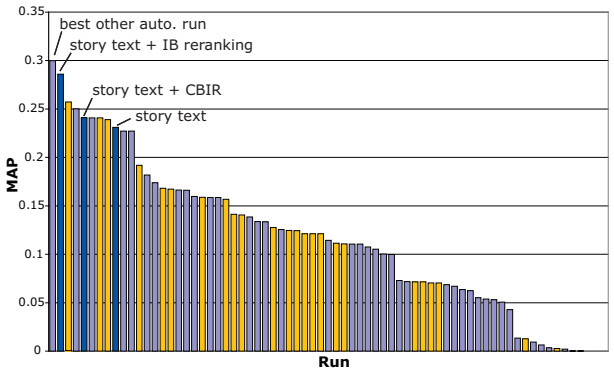

Figure 8: MAP on the six named-person queries of all automatic (blue) and manual (orange) runs in TRECVID 2005.

classes of queries. One simple example of such a model is the use of IB reranking when a named person is detected[6] in the query and a high-performing content-based image retrieval (CBIR) when no named entity is detected. We have experimented with this approach on the 24 queries in the TRECVID 2005 data set, using IB for the six named person queries and CBIR for the remaining 18 queries. Each approach has a MAP of about .11 across all 24 topics; however, using such a simple class-dependent fusion results in a jump in MAP to over 0.176, which outperforms the top manual submission (MAP of .168) as well as the top automatic run (MAP of .125).

7. CONCLUSION AND FUTURE WORK

We proposed a novel reranking process for video search, which requires no image search examples and is based on a rigorous IB principle. Evaluated on TRECVID 2003-2005 data set, the approach boosts the text search baseline over different topics in terms of average performance by up to 23%. In the future work, we will extend this method to incorporate visual appearance coherence so that the IB clusters not only preserve information about search relevance, but also maintain high visual consistency. Beyond the low-level features, we are extending the same framework on different feature representations such as mid-level concepts.

8. ACKNOWLEDGMENTS

We thank Eric Zavesky and Akira Yanagawa for supporting the video search platform. This material is based upon work funded in whole by the U.S. Government. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

9. REFERENCES

[1] A. Amir *et al.* IBM Research TRECVID-2005 video retrieval system. In *TRECVID Workshop*, Waishington DC, 2005.

[2] Alias-i. Lingpipe named entity tagger. In *http://www.alias-i.com/lingpipe/*.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[4] K. M. Donald and A. F. Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *CIVR*, Singapore, 2005.

[5] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *CVPR*, May 2004.

[6] S. Gordon, H. Greenspan, and J. Goldberger. Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations. In *ICCV*, 2003.

[7] A. G. Hauptmann and M. G. Christel. Successful approaches in the trec video retrieval evaluations. In *ACM Multimedia 2004*, New York, 2004.

[8] W. H. Hsu and S.-F. Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation. In *CIVR*, Singapore, 2005.

[9] W. H. Hsu and S.-F. Chang. Topic tracking across broadcast news videos with visual duplicates and semantic concepts. In *ICIP*, Atlanta, GA, USA, 2006.

[10] J. G. Carbonell *et al.* Translingual information retrieval: A comparative evaluation. In *International Joint Conference on Artificial Intelligence*, 1997.

[11] L. Kennedy, P. Natsev, and S.-F. Chang. Automatic discovery of query class dependent models for multimodal search. In *ACM Multimedia*, Singapore, November 2005.

[12] H. Liu. Montylingua: An end-to-end natural language processor with common sense. In *http://web.media.mit.edu/∼hugo/montylingua*.

[13] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *SIGIR*, Sheffield, South Yorkshire, UK, 2004.

[14] N. Slonim *et al.* Unsupervised document classification using sequential information maximization. In *25th ACM int. Conf. on Research and Development of Information Retireval*, 2002.

[15] A. Natsev, M. R. Naphade, and J. Tesic. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM Multimedia*, pages 598–607, Singapore, 2005.

[16] S. E. Robertson *et al.* Okapi at TREC4. In *Text REtrieval Conference*, pages 21–30, 1992.

[17] S.-F. Chang *et al.* Columbia University TRECVID 2005 video search and high-level feature extraction. In *TRECVID Workshop*, Waishington DC, 2005.

[18] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, 1992.

[19] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, 1999.

[20] T.-C. Chang *et al.* TRECVID 2004 search and feature extraction task by NUS PRIS. In *TRECVID Workshop*, Waishington DC, 2004.

[21] TRECVID. TREC Video Retrieval Evaluation. In *http://www-nlpir.nist.gov/projects/trecvid/*.

[22] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *CIVR*, Urbana-Champaign, IL, 2003.

[6] Automatic classification of query topics have been shown highly feasible for classes like named persons [11]