IBM Research TRECVID-2007 Video Retrieval System

Murray Campbell[*], Alexander Haubold[†], Ming Liu[‡], Apostol (Paul) Natsev[*],
John R. Smith[*], Jelena Tešić[*], Lexing Xie[*], Rong Yan[*], Jun Yang[§]

Abstract

In this paper, we describe the IBM Research system for indexing, analysis, and retrieval of video as applied to the TREC-2007 video retrieval benchmark. This year, focus of the system improvement was on cross-domain learning, automation, scalability, and interactive search.

Keywords—*Multimedia indexing, content-based retrieval, Support Vector Machines, Model Vectors, Model-based reranking.*

1 Introduction

We participated in the TREC Video Retrieval Track and submitted results for the High-Level Feature Detection and Search tasks. In this paper, we describe the IBM Research system and examine the approaches and results for both tasks. The video content is analyzed in an off-line process that involves audio-visual feature extraction, clustering, statistical modeling and concept detection, as well as speech indexing. The basic unit of indexing and retrieval is a video shot.

For the High-Level Feature Detection task, the IBM team continued its focus on integrating previously built assets into a highly automated, end-to-end detection system for high-level feature modeling. This system is designed to be scalable, configurable and extensible so that a large number of features can be detected with limited computational resources in a flexible learning process. The system has built-in a number of learning, normalization, sampling, parameter search, fusion and evaluation strategies, which greatly facilitate the effort on developing high-level feature detection algorithms, especially for naive users.

We also investigated several new learning algorithms along three directions: efficiency, cross-domain detection, and cross-concept detection. To significantly reduce learning and prediction time without degrading detection performance, we applied and evaluated a learning algorithm called random subspace bagging, which fuses an ensemble of SVM classifiers learned on randomly selected feature subspace and bootstrapped data samples. To utilize information from multiple domains, the training data from news video domain (TRECVID'05) was reused to augment the detection accuracy on TRECVID'07. Promising results have been observed. We also investigated the benefits of using a larger multimedia ontology to help improve the detection of individual concepts. We have used a rich LSCOM lexicon, coupled with simple combination strategies such as naive Bayes and logistic regression.

The highlight of IBM's search task this year was a new interactive search system designed to optimize manual annotation efficiency. The new system employs a hybrid tagging/browsing based annotation approach and switches to the most efficient annotation method based on estimated concept/topic selectivity and user annotation efficiency. To this end, two formal annotation models were developed to track and estimate annotation time for each user/topic pair. Based on the parameters of this model, the system merges the tagging-based and browsing-based annotation in order to minimize overall annotation time across the entire corpus and the full set of annotation concepts/topics. This hybrid annotation-based approach was applied to the interactive search task, resulting in the highest Mean Average Precision among all participants.

In addition to interactive search, the IBM team contin-

[*]IBM T. J. Watson Research Center, Hawthorne, NY, USA
[†]Dept. of Computer Science, Columbia University
[‡]Dept. of Computer Science, UIUC
[§]School of Computer Science, Carnegie Mellon University

ued its effort on automatic search, submitting 2 required baselines (visual-only and speech-only) and 3 optional automatic runs (2 type A and 1 type C). The two baselines performed competitively, ranking 4th and 7th respectively among the 25 submitted baselines, and a query-independent combination resulted in a small improvement over the baselines. Our query-dependent fusion approach did not generalize, however, leading to a slight loss in performance, most likely due to the changed relative importance of retrieval experts as compared to previous years' data sets and topics. Overall, our main emphasis this year was on expanding the concept lexicon for concept-based retrieval purposes, as well as on leveraging external resources to improve cross-domain robustness. In particular, we leveraged external annotations for about 50 generic concepts trained on consumer data (e.g., photos). We also used a large sample of web pages to re-estimate word frequencies in order to improve WordNet similarity measures based on information content. The external resources were used in a type C run, which was our best automatic run and performed approximately 30% better than the corresponding type A baseline.

2 Video Descriptors

2.1 Visual Features

Since the properties of TREC'07 video collection is significantly different from the video provided before, we performed extensive experiments on the development data in order to select the best feature types and granularities. We used a set of different visual descriptors at various granularities for each representative keyframe of the video shots. The relative performance of the specific features within a given feature modality (e.g., color histogram vs color correlogram) is shown to be consistent across all concepts/topics, but the relative importance of one feature modality vs. another may change from one concept to the other.

The following descriptors had the top overall performance for both search and concept modeling experiments:

- Color Histogram (CH)—global color represented as a 166-dimensional histogram in HSV color space.
- Color Correlogram (CC) — global color and structure represented as a 166-dimensional single-banded auto-correlogram in HSV space using 8 radii depths [HKM+99].
- Color Moments (CMG) — localized color extracted from a 5x5 grid and represented by the first 3 moments for each grid region in Lab color space as a normalized 225-dimensional vector.
- Co-occurrence Texture (CT)—global texture represented as a normalized 96-dimensional vector of entropy, energy, contrast, and homogeneity extracted from the image gray-scale co-occurrence matrix at 24 orientations.
- Wavelet Texture Grid (WTG)—localized texture extracted from a 3x3 grid and represented by the normalized 108-dimensional vector of the normalized variances in 12 Haar wavelet sub-bands for each grid region.
- Edge Histogram (EH)—global edge histograms with 8 edge direction bins and 8 edge magnitude bins, based on a Sobel filter (64-dimensional).

We used cross-validation to optimize the fusion strategy for each concept individually in high-level feature task. For the search task, we used the descriptors that have consistent top performance for both concept detection and search experiments. We use the term *visual-based approach* to denote search methods in low-level visual descriptor space.

2.2 HoG Features

We introduce a novel low-level visual features provided by UIUC called the locally normalized Histogram of Oriented Gradient (HOG). Originally, these features are designed for robust human detection task and provide excellent performance relative to other existing feature sets including wavelets. They are reminiscent of edge orientation histograms, SIFT descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalization to improve performance.

The entropy, as the description length of a random variable, has a very nice property in characterizing the color

distributions. The original HOG feature is a 3780 dimensional vector for the entire image. We construct our HOG feature by the following steps:

1. For each image, converted it from RGB color space to HSV color space;
2. Build a normalized histogram in the H (hue) channel;
3. Compute the entropy of the normalized color histogram in H channel;
4. Concatenate the entropy (a real number) to the 3780 HOG feature to form a 3781 dimensional vector.

The rest part of the detector is just like the traditional HOG detector. Please refer to [DT05] for details.

2.3 Semantic Features

The Large-Scale Concept Ontology for Multimedia (LSCOM) is designed to simultaneously optimize utility, facilitate end-user access, cover a large semantic space, make automated extraction feasible, and increase observability in diverse broadcast news video data sets[NST+06]. Some of the semantic-based retrieval and high-level feature detection approaches presented in this work rely on a previously modeled high-level semantic feature space, including both the LSCOM concepts and the LSCOM-lite concepts. For cross-concept detection, we leverage these ontologies to help improve the detection of individual concepts.

For semantic-based retrieval, we apply concept detection to query examples and generate model vector features consisting of the confidences of detection for each of the concept models in our lexicon [NNS04], as following:

- LSCOM-lite – 39-dimensional vector consisting of concept scores from the full LSCOM taxonomy and was jointly annotated by the TRECVID community in 2005; subset of 36 relevant concept was annotated on TRECVID 2007 development set.
- LSCOM – 155-dimensional vector consisting of concept scores of top performing 155 concepts from full LSCOM lexicon.
- Consumer – 50-dimensional vector consisting of concept scores of 50 models trained on consumer and web images.

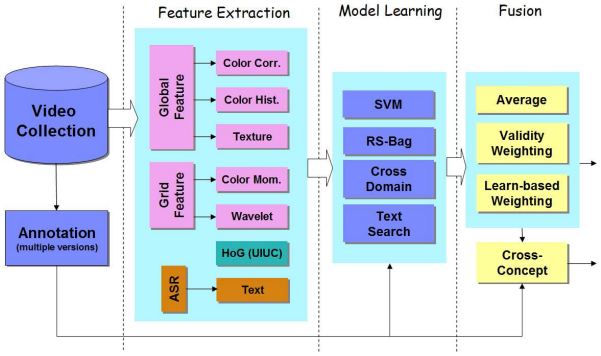

Figure 1: Overview of the IBM 2007 TRECVID High-level Feature Detection System.

3 High-level Feature Detection

Figure 1 illustrates the IBM high level feature detection system. Our basic principle for modeling high-level features has consistently been to apply supervised learning algorithm to low-level features [NNT05] as described in Section 2. The criterion has always been to leverage generic learning algorithms for all concepts rather than focus on an overly specific and narrow approach that can only work for a single concept. In our view, generic learning provides the only scalable solution for learning the large scale semantics needed for efficient and rich semantic search and indexing. Our current system includes multiple base and meta-level learning algorithms such as SVMs, random subspace bagging, cross-domain learning and so on. It also consists of different fusion strategies and cross-concept learning components for leveraging multi-modal and multi-concept relationship. The details of these components are explained in details in the rest of this section.

3.1 General Approaches and Modeling Tool

As the baseline approaches, we re-implement the same general learning algorithms that have been proven in the past to be successful, and switch our focus to integrate previously built assets into a highly automated, end-to-end detection system for high-level feature detection. In the training stage, low-level feature representations are learned corresponding to the binary annotations for each

concept using support vector machines (SVMs). This is because SVMs have resulted in top performance in the task of high-level feature extraction for previous NIST TRECVID evaluations. In particular, we use non-linear kernels for the global-based and grid-based visual features, and use linear kernels for the HoG features. This year two shared versions of annotations are officially recommended by NIST, where one is from the collaborative annotation forum organized by LIG [AQ07] and the other is provided by the MCQ-ICT-CAS team. In view of the noticeable difference between these two annotations, we compute both their union and intersection to serve as the baseline relevant judgment.

Performance of SVM classifiers can vary significantly with variations in model parameters. Choice of learning parameters is thus crucial for the results. To minimize sensitivity of the choices of parameters, we choose the parameters based on a grid search strategy. In our experiments, we build models for different values of the RBF kernel parameters, the relative cost factors of positive vs. negative examples, feature normalization schemes, and the weights between training error and margin. The optimal learning parameters are selected based on the average precision using 2-fold cross validation on development data. Before the learning process, the distribution between positive and negative data are re-balanced by randomly down-sampling the negative data to a smaller size. For each low-level feature, we select one optimal configuration to generate the concept model. Finally, four best-performed models are combined to be a composite classifier by averaging. In the detection stage, we apply the optimal model to evaluate the target images for the presence/absence of the concepts, and generate a confidence measure that can be used to rank the testing images.

Figure 2 illustrates the IBM VClass modeling tool for modeling and optimizing the high-level semantic features. This system is designed to be scalable, configurable and extensible so that a large number of features can be detected with limited computational resources in a flexible learning process. It automates a number of critical steps for concept detection, including concept learning, feature normalization, data/feature sampling, parameter search, multimodal fusion and performance evaluation, which creates a simple interface for non-experts who want to build good quality models based on several best practices that we have developed over the past six years. Although

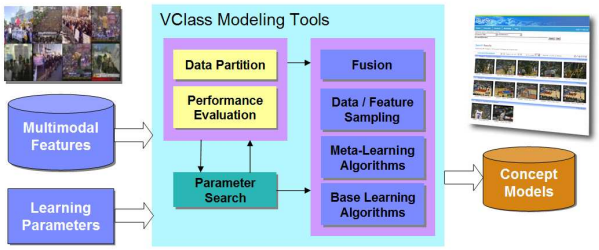

Figure 2: IBM VClass modeling tools for building high-level concept models. The system contains a number of components for concept learning, feature normalization, data/feature sampling, parameter search, multimodal fusion and performance evaluation.

this fully automated process might result in a slightly inferior performance to the manually tuned learning process, manual effort spending in developing new concept detection systems are significantly reduced. Thus it leads to a more efficient process when migrating systems to other types of data collections and high-level features, which is critical for the visual-based concept detection systems.

For the text-based concept detection, we leveraged our speech-based retrieval system (see Section 4.1) to generate search-based results for each concept. Specifically, we manually created text-based queries for each concept based on interactive search results on the development set. We used the automatic query expansion and word suggestion capabilities of our interactive system to identify relevant keywords for each concept and to optimize the performance of the retrieval system. The queries were then applied automatically on the test set and the search results were used as our text-based concept detection run.

3.2 Scalability

In reality, multimedia data collections can contain hundred thousands to millions of items, and these items can be associated with thousands of different labels. However, most existing algorithms do not scale well to such a high computational demand. For example, most current support vector machine(SVM) implementations have a learning time of $t_l(n,m) = O(mn^2)$ and a prediction time of $t_p(n,m) = O(mn)$ where m is the feature size and n is the data size. Therefore, the computational resources

Algorithm 1 The round-robin random subspace bagging (RSBag) algorithm for high-level feature detection.

Input: training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1...N}$, $\mathbf{x}_i \in \mathbb{R}^M$, number of total base models T, number of labels L, data sampling ratio $r_d(\leq 1)$, feature sampling ratio $r_f(\leq 1)$.

1. For t = 1 to T,
   (a) Choose a label $l = \mu(t)$, where $\mu(\cdot)$ is the remainder over L;
   (b) Take a bootstrap sample X_+^t from positive data $\{\mathbf{x}_i\}$ for l (i.e., $y_{il} = 1$), $|X_+^t| = Nr_d/2$;
   (c) Take a bootstrap sample X_-^t from negative data $\{\mathbf{x}_i\}$ for l (i.e., $y_{il} = -1$), $|X_-^t| = Nr_d/2$;
   (d) Take a random sample F^t from the feature indices $\{1, ..., M\}$, $|F^t| = Mr_f$;
   (e) Learn a base model $h^t(\mathbf{x})$ using X_+^t, X_-^t and F^t. SVMs are used in this work.
   (f) $F_l(\mathbf{x}) \leftarrow F_l(\mathbf{x}) + h^t(\mathbf{x})$.
2. Output the classifier $y_l = sign[F_l(\mathbf{x})]$.

needed to learn millions of data will be prohibitive even after negative data are down sampled. To speed up high-level feature extraction without performance degradation, one approach is to exploit the information redundancy in the learning space. To this end, researchers have proposed several ensemble learning algorithms based on random feature selection and data bootstrapping. Combining bagging (bootstrap aggregation) and RSM, Breiman has developed a more general algorithm called *random forest* [Bre01]. Random forest aims to aggregate an ensemble of unpruned classification/regression trees using both bootstrapped training examples and random feature selection in the tree induction process. Random forest can be learned more efficiently than the baseline method, and it has empirically demonstrated superiority compared to a single tree classifier.

However, ensemble learning approaches were not limited to tree classifiers. The extended idea of bagging was applied in a video retrieval task [NNT05], and the random forest idea was used in an image retrieval task [TTLW06]. In general, we term the algorithmic combination of bagging and random subspace selection as "random subspace bagging" (RSBag) classifiers (a.k.a., Asymmetric Bagging and Random Subspace classifiers in previous work [TTLW06]). To apply the random subspace bagging algorithms for the multi-label classification problem, we present a round-robin random subspace bagging approach in Algorithm 1. This algorithm first selects a label to work with in a round robin fashion. Then we learn a base model based on Nr_d balanced bootstrap samples from the positive and the negative data, together with Mr_f random samples from the feature indices, where r_d is called the *data sampling ratio* and r_f is called the *feature sampling ratio*. Both sampling ratios are determined by the input parameters and typically they are less than 1. At the end we aggregate all the base models for the same label into a composite classifier containing T base models. This algorithm is similar to a balanced version of random forests [TTLW06]. The only difference is that it can use any binary classifier to construct the base models without being limited to the decision trees.

In this work, SVMs serve as the base models with the number of models as 3, a feature sampling ratio r_f as 0.5 and a data sampling ratio r_d selected from {0.1,0.2} based on 2-fold cross validation. Based on the theoretical computation of its time complexity, we will be able to achieve a 16 to 67 fold speedup for training, and a 3 to 7 fold speedup for prediction. As future work, we also plan to explore the effectiveness of our recent proposed method called "model-shared subspace boosting" [YTS07], which aims to reduce the information redundancy across multiple concepts.

3.3 Cross-domain Ensemble Detection

To improve high-level feature extraction in TRECVID 2007, we can exploit additional training data from other domains. A readily available source of data is the development set of TRECVID 2005, which have been manually labeled with respect to the 39 LSCOM-Lite high-level features. Although TRECVID 2005 data are broadcast news footage while TRECVID 2007 data are magazine video, which may have different distributions, using the additional training data is still beneficial in two cases. First, some features are very infrequent and the number of positive instances in the development set of TRECVID 2007 is too small to build a reliable classifier. Second, features such as "outdoor" and "car" are generic ones and

their underlying distribution is insensitive to the change of domains. In both cases, out-of-domain training data are likely to be helpful.

We adopt an ensemble approach to use the training data in TRECVID 2005. For each high-level feature, we train a classifier from the TRECVID 2005 data based on the best parameters found by cross-validation. Meanwhile, we have a classifier for the same feature built from TRECVID 2007 data. The output (i.e., prediction scores) of the two separate classifiers are combined in the form of a weighted sum, where the weights are set proportional to their classification performance measured by average precision (AP) on the development set of TRECVID 2007. Since the second classifier is trained from TRECVID 2007, its AP is calculated based on cross-validation.

3.4 Cross-concept Detection

To leverage the context between multiple concepts, we first explore the pair-wise correlations between the target c and each concept i in the lexicon M. Taking as input the detection scores of all concepts $y = [y_1, \ldots, y_i, \ldots, y_M]^T$, we obtain the maximum-likelihood estimate the pair-wise conditional probabilities $P(y_i|y_c)$. We then use these estimates in a Naive Bayes model (Eq. (1)) to obtain the cross-concept log-likelihood ratio L_c (Eq. (2)).

$$P(y_c|y_{1:M}) \propto P(y_c) \prod_{i=1:M} P(y_i|y_c) \quad (1)$$

$$L_c = \log \frac{P(y_c=1|y_{1:M})}{P(y_c=0|y_{1:M})} \quad (2)$$

$$= \log \frac{P(y_c=1)}{P(y_c=0)} + \sum_{i=1:M} \log \frac{P(y_i|y_c=1)}{P(y_i|y_c=0)}$$

For TRECVID'07 target concept c, we use the rest of 35 LSCOM-lite target concepts, along with a selected subset of 157 LSCOM concepts (trained on the 2005 development corpus) as the input concept pool. We partitioned the TRECVID'07 development set into two, learned a set of ensemble-SVM detection scores from one half and obtained model parameters $P(y_i|y_c)$ on the other half. Each input dimension y_i is sigmoid-normalized and uniformly quantized into 15 bins, and the maximum likelihood estimates of $P(y_i|y_c)$ is smoothed by the prior of y_i as $\hat{P}(y_i|y_c) = (1-\alpha)P(y_i|y_c) + \alpha P(y_i)$, with $\alpha = 0.06$. The resulting likelihood scores L_c are then used to re-rank the original prediction score with averaging.

3.5 Fusion Methods

We applied ensemble fusion methods to combine all concept detection hypotheses generated by different modeling techniques or different features. In particular, we performed a grid search in the fusion parameter space to select the optimal fusion configuration based on a repartition of development data. Fusion parameters include a score normalization method and a score aggregation method. For score normalization methods, we considers using both raw SVM scores and sigmoid normalization. The fusion methods we considered include simple average and weighted average fusion. As a special case of weighted averaging, we considered validity-based weighting, where the weights are proportional to the average precision performance of each concept detection hypothesis on a held-out validation set. We also considered learning-based weighting, where the weights are learned by a meta-level SVM classifier. The fusion strategy are automatically chosen based on the validation performance.

To generate the runs, we detect the concepts first using the following individual approaches and then proceeded to fuse resultant retrieval lists with described fusion techniques. By default, we use the union of both official groundtruth annotations to learn the concept models unless stated otherwise.

1. SVM-07: SVM Models built for TRECVID 2007 using IBM VClass modeling tool on low-level visual features;

2. SVM-Min07: SVM Models built for TRECVID 2007 using IBM VClass modeling tool on low-level visual features using the intersection of both official groundtruth annotations;

3. SVM-HoG: SVM Models built on the HoG features;

4. SVM-05: SVM Models built on TRECVID 2005 development data for all 39 concepts;

5. Text: Text retrieval with manually defined keywords for each concept;

Description	Run	Type	MAP	Time
SVM-07	-	-	0.0638	24602
SVM-Min07	-	-	0.0600	21460
Fusion of SVM-07 and SVM-Min07	Max.Min	A	0.0667	-
Fusion of SVM-07, SVM-Min07 and Text	Max.Min.Text	A	0.0797	-
Fusion of SVM-07 and SVM-05	Max.LSCOM	A	0.0729	-
Fusion of SVM-07, HoG and Text	Max.HoG.Text	A	0.0784	-
Fusion of SVM-07, SVM-Min07, Text and HoG	-	-	0.0844	-
Fusion of SVM-07, SVM-Min07, Text, HoG and SVM-05	-	-	0.0930	-
RSBag	-	-	0.0679	2342
Fusion of RSBag, SVM-Min07, Text, HoG and SVM-05	-	-	0.0953	-
Cross concept on SVM-07 + SVM-05	CrossConcept	A	0.0781	-
Cross concept on SVM-07 + SVM-Min07 + Text + HoG + SVM-05	-	-	0.0959	-
Cross concept on RSBag + SVM-Min07 + Text + HoG + SVM-05	-	-	0.0975	-

Table 1: IBM TRECVID 2007 High level Feature Detection Task – Submitted and Unsubmitted Runs

6. RSBag: Random subspace bagging with SVM base models;

7. LSCOM: To leverage a larger scale of context, we built additional concept models using the entire LSCOM annotations on TRECVID 2005 data.

3.6 Submitted Systems and Results

We have generated multiple runs of detection results based on the approaches presented before. A number of runs are submitted to NIST for official evaluation with their submission name shown, and all of the remaining runs are evaluated using the ground truth provided by NIST. The mean inferred average precision is used as the measure of the overall performance of the systems. Table 1 lists the performance of all the submitted and unsubmitted runs, and Figure 3 summarizes the retrieval performance of IBM high level feature runs in context of all the Type A submissions. The baseline runs, i.e., SVM-07 and SVM-Min07, achieve similar performance as each other, and combining both of them provides a 5% improvement over SVM-07. By introducing complementary information beyond visual features, text retrieval results in another 20% improvement over visual-only detection. After the HoG features are incorporated, the MAP are further increased from 0.0784 to 0.0844. Development data from the news video domain are also proven to be informative,

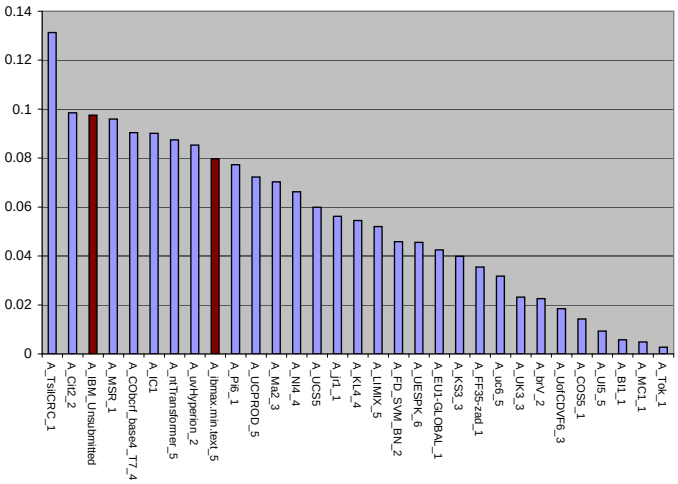

Figure 3: Retrieval performance of IBM high level feature runs in context of all the Type A submissions using the mean inferred average precision measure. Due to space limit, only the best run of each organization are shown.

because fusing the training data from TREC'05 can improve the performance by another 10%. Finally, we find that cross-concept reranking brings consistent improvement in MAP over runs of all flavors. Moreover, we also observe improvement in almost all 20 evaluated concepts.

Most concepts can benefit from multi-modal fusion, but the usefulness of each modality to individual concepts

may vary. For example, text features are most useful when detecting the concepts of "Airplane" and "Boat_Ship", while training data from news video domain are most useful when detecting the concepts of "Waterscape" and "Maps". Along another direction, we also list the training time for SVM-07, SVM-Min07 and RSBag in order to compare their computational efficiency. As can be seen, RSBag provides a more than 10-fold speedup on the training process and even a slightly better performance than the baseline SVM-07 and SVM-Min07 methods. Note that, this speedup is less than the theoretical prediction, because additional I/O time for reading/writing data are also taken into consideration in our measurement. However, these results clearly demonstrated both the effectiveness and efficiency of the random-subspace sampling method. Combining RSBag with other learning components results in our best run with a MAP of 0.975.

4 Automatic Search

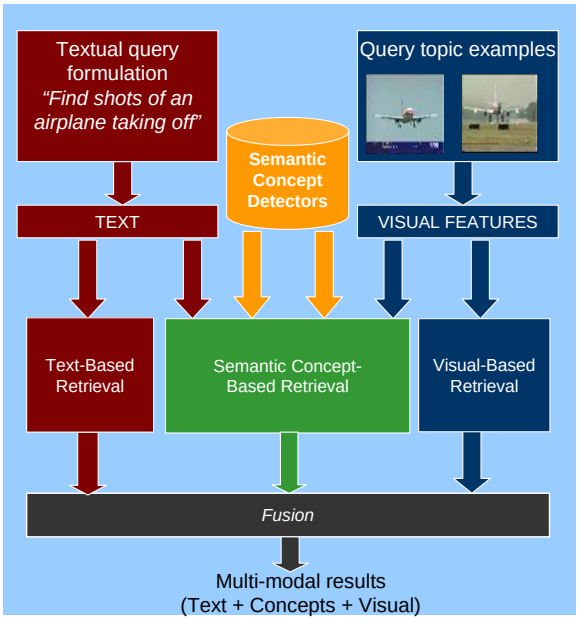

Figure 4: Overview of IBM automatic search system.

The IBM team continued its effort on automatic search, submitting 5 automatic runs (4 type A runs and 1 type C run). The overall architecture of our automatic search system was again a combination of speech-based retrieval with automatic query refinement, model-based query retrieval and re-ranking based on automatic mapping of textual queries to semantic concept detectors, semantic retrieval based on automatic mapping of visual query examples to concept detectors, and visual retrieval based on light-weight learning and smart pseudo-negative sampling (see system overview in Figure 4). All processing was done at the sub-shot level based on the master shot boundary reference [Pet], where each sub-shot was represented by a single keyframe and a corresponding speech transcript segment. All ranking results were generated at the sub-shot level first and then aggregated at the shot level by taking the maximum confidence score across all sub-shots for each master shot.

This year we again emphasised concept-based retrieval and re-ranking by leveraging a larger set of automatic concept detectors. In addition to the 36 concept detectors we built for the High Level Feature Detection task, we also trained detectors for a subset of 155 concepts from the full LSCOM set. For our type-C run, we also used an additional 50 concept detectors trained on external consumer photo collections. We experimented with several approaches for automatic query-to-model mapping and weighting, including a lexical WordNet-based approach mapping query text to concepts, a statistical co-occurrence-based approach mapping query text to concepts, as well as a statistical content-based approach mapping visual query examples to concepts. These concept-based retrieval results were not submitted as standalone runs but were combined with speech-based and/or visual-based retrieval baselines in order to generate concept-based expanded and re-ranked results.

4.1 Speech-based retrieval

Our speech-based retrieval system is the same as last year and is based on the JuruXML semantic search engine [MMA+02]. In particular, we indexed the ASR/MT transcripts corresponding to each sub-shot from the master shot reference provided by Fraunhofer (Heinrich Hertz) Institute in Berlin [Pet]. For the purposes of speech transcript indexing, each sub-shot was first expanded to include several neighboring sub-shots in order to account for mis-alignment between the spoken and visual tracks. The corresponding transcript falling within the expanded

shot boundaries was then indexed as a pseudo text document representing the given shot.

At retrieval time, we leveraged the native query expansion functionality of the JuruXML search engine to automatically refine the query based on pseudo-relevance feedback and Lexical Affinities, or pairs of words that tend to co-occur in close proximity of each other (e.g., phrases) [CFPS02]. Parameters of this query refinement approach included the number of top documents to consider (pseudo-)relevant, the max number of new query terms to add, the weight of the newly added query terms, and the weight of lexical affinities relative to single keywords. All of these parameters were tuned empirically using the development set and the common annotation ground truth for the 36 LSCOM-lite concepts. In particular, we manually created text queries to match each of the 36 concepts, and we used Mean Average Precision on the development set to compare and tune indexing and query expansion parameters. The resulting speech-only baseline had a MAP score of 0.015 on the test set.

4.2 Concept-based retrieval

Concept-based retrieval applies the results from off-line concept detection and text analysis to on-line queries by triggering concept models with different weights. Given an arbitrary text query, the goal is to identify which concepts, if any, are relevant to the query, and to what extent (i.e., what should the weights for each concept be in a weighted fusion scheme). Once the final list of most relevant concept models and weights are determined, we fuse the corresponding concept detection result lists using weighted average score aggregation to generate a final ranked list of shots. This concept-based query result list is then used to re-rank results generated from other retrieval methods through an appropriate fusion method. For concept-based retrieval purposes we used our 36 detectors from the High Level Feature Extraction task, as well as detectors we built for 155 LSCOM concepts [NST+06] chosen primarily based on their frequency in the training data. For our type C run, we also used approximately 50 generic and highly robust concept detectors trained on external consumer photo collections. For both type A and type C runs, we used light-weight query topic modeling in order to map the visual query examples to relevant concepts and weights, as described in Section 4.3. For more details, see [TNS07].

For our type A runs, we considered a statistical co-occurrence-based approach, which identifies co-occurrence relationships between words from the speech transcript and visual concept labels. The approach use statistical hypothesis testing based on the G^2 score to identify only the significant such relationships, which are then used for query expansion purposes. For more details, see [NHT+07].

For our type C run, we also considered a WordNet-based lexical query expansion approach, which used the Jiang-Conrath similarity measure to identify soft matches between query terms and concept synonyms. This approach was essentially the same as [HN06] but using Jiang-Conrath instead of Lesk similarity. Also, since the Jiang-Conrath similarity measure is based on the information content of words (which is derived from their frequency), we used a large sample of web pages to re-estimate the information content of words present in WordNet. We found this to be necessary since the original Brown corpus, which is traditionally used to compute information content, is quite outdated and does not contain over 70% of the words present in WordNet today. The use of external web pages is the reason for flagging this run as type C.

4.3 Visual-based Retrieval

Two components of IBM TRECVID automatic search system rely solely on query topic visual examples. We focus here on a visual content-based approach, where the queries are comprised of one or more visual examples, as opposed to textual keywords. Thus, the underlying retrieval approach is essentially the same for both components, and formulates the topic answering problem as a discriminant modeling one. SVM modeling with nonlinear kernels allows us to learn nonlinear decision boundaries even when the descriptors are high dimensional. We fix the kernel type to Radial Basis Kernels, and select global SVM kernel parameters for each descriptor to avoid over-fitting. Since there are no negative examples provided, we generate pseudo-negative examples by randomly sampling data points. We build multiple primitive SVM classifiers whereby the positive examples are used commonly across all classifiers but each has a different sampled set of pseudo-negative data points. The

SVM scores corresponding to all primitive SVM models are then fused using AND logic to obtain a final discriminative model.

Our improvements this year over baseline method include the intelligent modeling of the positive and pseudo-negative space using unbiased and biased methods for data sampling and data selection. We apply the proposed method in a fusion framework to improve discriminative support vector machine modeling, and improve the overall system performance. The result is an enhanced performance over any of the baseline models, as described in [TNXS07].

4.4 Multimodal Fusion and Reranking

The final component of the IBM automatic search system is multimodal fusion. We have used query-dependent search fusion among the text, model, semantic and visual retrieval scores. This fusion processing involve two steps: extracting query features and mapping test queries to known ones and/or learning optimal combination weights over known queries. A detailed description can be found in our prior paper [XNT07].

Query features are generated using the input query text. We use the PIQUANT [CCCP$^+$04] engine to tag the query text with more than one hundred semantic tags in a broad ontology, designed for question-answering applications on intelligence and news domains. The set of tags cover person, geographic entities, objects, actions, events, etc. For instance, "Hu Jintao, president of the People's Republic of China" would be tagged with "Named-person, President, Geo-political Entity, Nation". Note that in this example, multiple annotations lead to the same visual meaning ("Named-person, President" $\rightarrow$ person), while some annotations may not have direct visual implications ("Nation"). Hence a *de-noising* step is needed to map these annotations into a few distinct visual categories. We design such a mapping manually from all semantic tags to seven binary feature dimensions, intuitively described as *Sports, Named-Person, Unamed-Person, Vehicle, Event, Scene, Others*. This mapping consists of a few dozen rules based either on commonsense ontological relationship, e.g., a semantic annotation "Person:NAME" leads to *NamedPerson*, or on frequent co-occurrence, such as "Road" implies *Vehicle*.

We use dynamically generated query weights. Each new query is mapped to a small set of *neighbors* among the training queries (with inner product of query features serving as the similarity measure), and the weights for the unseen query are obtained by maximizing the average performance on the current set of neighbors. Nearest neighbor mapping with exhaustive search on the performance space works well in this case, since the query feature space here is rather low-dimensional.

4.5 Experiments and Results

We submitted 5 automatic runs (4 type A and 1 type C) for this year's Search Task, which are listed with their corresponding MAP scores in Table 2.

For our required speech-only retrieval baseline (run TJW_Text), we used the common ASR/MT transcripts in English, and the baseline retrieval system had a MAP score of 0.015. Despite of the low absolute MAP score, this appears to be one of the top performing speech-only baselines. We attribute the low absolute score on the lack of named entity topics and the poor quality of the ASR/MT transcripts.

Our visual retrieval system was again based on light-weight learning and query topic modeling based on Support Vector Machines, combined with smart pseudo-negative sampling and bagging approaches. This system generated our highest MAP score (0.03) but the performance was unfortunately somewhat deteriorated after fusion with the speech-based and/or model-based retrieval results. In particular, after simple averaging of the visual retrieval results and the concept-based retrieval results, our visual retrieval baseline (run TJW_MSV) produced a MAP score of 0.022—a loss of nearly 30% with respect to the original visual-only run.

The multi-modal fusion of the speech-based, visual-based, and concept-based results produced two runs—TJW_TMSV.qind and TJW_TMSV.qdyn. The former—query-independent—run was generated by simple averaging, while the latter—query-dynamic—run was generated using dynamically-generated query-specific fusion weights [XNT07]. Unfortunately, the query-dependent fusion was trained on queries and data from TRECVID 2005 and 2006, where performance of the individual retrieval experts was quite different as compared to 2007. Because of this, the fusion weights did not generalize, leading to worse performance than the query-independent

Run ID	Run Description	Run Type	Run MAP
F_A_1_TJW_Text_6	Speech-only baseline	A	0.0151
F_A_1_TJW_MSV_5	Visual-only baseline	A	0.0215
F_A_2_TJW_TMSV.Qind_4	Multi-modal fusion (query independent)	A	0.0233
F_A_2_TJW_TMSV.Qdyn_2	Multi-modal fusion (query dynamic)	A	0.0212
F_A_2_TJW_TMSV-C.Qdyn_3	Type-C multi-modal fusion (query dynamic)	C	**0.0275**

Table 2: Mean Average Precision scores for all IBM automatic search submissions.

fusion. Specifically, run TJW_TMSV.qind generated a MAP score of 0.023, while run TJW_TMSV.qdyn generated a MAP score of 0.021.

Our final run, TJW_TMSV-C.qdyn, was similar to the query-dependent multi-modal TJW_TMSV.qdyn run, with the exception that it included about 50 additional concept detectors trained on external data, as well as a type-C concept-based retrieval run, which leveraged a large sample of web pages to estimate the information content of words in WordNet (see Section 4.2). This run produced a MAP score of 0.028, which was an improvement of about 30% with respect to the corresponding type-A run, TJW_TMSV.qdyn.

Overall, our emphasis in this year's experiments with automatic search was on expanding the concept vocabulary for concept-based retrieval and re-ranking purposes by leveraging cross-domain concept detectors. The results were mixed, however. On the one hand, the use of a larger set of LSCOM models did not seem to improve retrieval performance significantly. We believe this is most likely due to lower accuracy of the general LSCOM concept detectors (trained on a different domain), coupled with ineffective multi-modal fusion (also trained on a different domain). On the other hand, the use of generic consumer concept detectors (which were typically quite robust) did help to significantly improve the performance of our type-C run (which was about 30% better than its corresponding baseline). This means that cross-domain concepts can be effective and quite useful in improving retrieval performance but it is critical that the models are robust and carefully selected. In the future, we plan to investigate strategies for selecting concept detectors with the most potential to help for cross-domain retrieval.

5 Interactive Search

In this section, we present a novel interactive retrieval system that has been demonstrated to be highly effective in this year's evaluation.

5.1 Annotation-based interactive retrieval

In general, the users of interactive retrieval can be categorized into three broad categories. (1) A general class of users aimed at *browsing* over a large number of videos from diversified sources, where users have no specific target at the beginning except for finding interesting things. (2) Another class of users want to do *arbitrary search* by retrieving an arbitrary video satisfying his information need that can be presented by text keywords or visual examples. Arbitrary search usually places more emphasis on precision in top-ranked clips. (3) The third class of users, *complete search/annotation*, aims to discover every relevant video that belong to a specific information need. To support these users, the retrieval systems must possess more automatic processing power to reduce the huge manual annotation efforts. In these systems, recall or average precision in the entire collection is an important criterion to optimize.

Clearly, the interactive retrieval task designed by TRECVID falls into the third category. In other words, it is most similar to a video annotation task which aims to annotate the entire video collection with some given topics. Therefore, in the following discussions, we would like to consider using manual image annotation approaches to address the interactive retrieval problem by treating the query topics as annotation keywords.

5.2 Manual Image Annotation and Their Formal Models

Recent years have seen a proliferation of manual image annotation systems for managing online/personal multimedia content. This rise of manual annotation partially stems from its high annotation quality for self-organization/retrieval purpose, and its social bookmarking functionality that allows public search and self-promotion in online communities. Manual image annotation approaches can be categorized into two types. The most common approach is *tagging*, which allows the users to annotate images with a chosen set of keywords ("tags") from a controlled or uncontrolled vocabulary. Another approach is *browsing*, which requires users to sequentially browse a group of images and judge their relevance to a pre-defined keyword. Both approaches have strengths and weaknesses, and in many ways they are complementary to each other. But their successes in various scenarios have demonstrated that it is possible to annotate a massive number of images by leveraging human power.

However, manual image annotation can be a tedious and labor-intensive process. Therefore, it is of great importance to consider using automatic techniques to speed up the manual image annotation process and help users to create more complete/diverse annotations in a given amount of time. Here we assume users will drive the annotation process and manually examine each image label in order to guarantee the annotation accuracy, but in addition we improve the annotation efficiency by automatically suggesting the right images, keywords and annotation interfaces to users. This is different from the automatic image annotation task, which aims to construct accurate visual models based on low-level visual features.

Until now, there are few studies available on quantitatively analyzing and optimizing the efficiency of the manual image annotation process. We attribute this to a lack of formal annotation time/efficiency models for evaluating large-scale manual annotation. Therefore we briefly describe two formal annotation time models for two popular single-user manual annotation approaches, i.e., tagging and browsing. More details can be found in our recent paper [YNC07].

5.2.1 Tagging

Tagging allows the users to annotate images with a chosen set of keywords("tags") from a controlled or uncontrolled vocabulary. This type of approaches is the basis for most of the current image annotation/tagging systems, such as Flickr [fli] and ESP Game [vAD04], although it can be implemented in a variety of ways with respect to interface designs and user incentives. One advantage for tagging is that annotators can use any keywords in the vocabulary or freely choose arbitrary words to annotate target images. However, this flexibility might result in a "vocabulary problem" [FLGD87], which means multiple users or a single user in a long period can come up with different words to describe the same concept. This vocabulary disagreement can lead to inefficient user interaction or missed information in the annotation process. Moreover, it can be more time-consuming for general users to provide new keywords, as compared with simply browsing and judging the relevance between image and a predefined keyword.

In order to quantitatively analyze the efficiency of tagging approaches, we must design a formal model to represent its annotation time for each image. To begin, we can assume that the more keywords users annotate, the larger the annotation time is. Based on the user study presented in [YNC07], we model the tagging time T_l for the l^{th} image as a function of four major factors, i.e., the number of image keywords K_l, the average time for designing/typing one word t_f, the initial setup time for annotation t_s and a noise term ϵ, which follows an zero-mean probability distribution in order to capture the variance for T_l. The user study suggests it is sufficient to adopt a linear time model to represent the annotation time for each image, i.e., $T_l = K_l t_f + t_s + \epsilon$. Its mean can be derived as $t_l = K_l t_f + t_s$. For a total of L images, the overall expected annotation time is

$$t = \sum_{l=1}^{L} K_l t_f + L t_s \quad or \quad t = \sum_{k=1}^{K} L_k t_f + L t_s. \qquad (3)$$

Note that this time model does not require the parameters t_e and t_f to be constant in all the annotation scenarios. Instead, they can be affected by a number of factors, such as interface design, input device, personal preference and so on. For example, annotation on cell phones can have

a much larger t_f than annotation on desktop computers. Therefore, rather than attempting to estimate fully accurate parameters for any specific settings, this paper mainly focus on examining the correctness of the proposed time models, and use them as a foundation to develop better manual annotation algorithms. We expect the proposed time models and the following analysis will generalize over a wide range of settings.

5.2.2 Browsing

Another type of annotation approach, *browsing*, requires users to browse a group of images, so as to judge the relevance of each image to a given keyword. The number of images per group can vary from 1 to a large number such as 20. Examples include Efficient Video Annotation (EVA) system [VSN05] and Extreme video retrieval (XVR) [HLY+06]. Because browsing annotation needs to start with a controlled vocabulary defined by domain experts or a seeded keyword manually initialized by users, it is not as flexible and as widely applied as tagging. However, browsing annotation has its own advantages on several aspects. For instance, it allows users to provide more complete annotation outputs than tagging [VSN05], because in this case users only focus on one specific keyword at a time and they do not need to remember all possible keywords over a long period. Moreover, the time to annotate one keyword by browsing is usually much shorter than that in tagging, since users have a relatively simple binary judgment interface and a stable annotation context in the entire process.

Similar to tagging, we design a formal annotation time model in order to quantify the efficiency for browsing. Since all the images have to be examined for each keyword, the overall annotation time should be related to the number of images and the number of unique keywords. According to the user study presented in [YNC07], we also find that the time for annotating a relevant image is significantly larger than the time for skipping an irrelevant image, because users tend to spend more time and be more careful on examining the correctness on relevant images. Based on these observations, we model the browsing annotation time T_k for the k^{th} keyword using four major factors, i.e., the number of relevant images L_k, the average time to annotate a relevant image t_p, the average time to annotate an irrelevant image t_n and a noise term ϵ which follows a zero-mean probability distribution. The number of irrelevant images is simply $\bar{L}_k = L - L_k$ and hence a reasonable linear time model is $T_k = L_k t_p + (L - L_k)t_n + \epsilon$. For a total of K keywords, the overall expected annotation time is

$$t = \sum_{k=1}^{K} \left[L_k t_p + (L - L_k) t_n \right]. \quad (4)$$

To summarize, these two annotation approaches are essentially complementary to each other from many perspectives. For example, tagging has less limitations on the choice of words and users only need to handle the relevant keywords for each image. But the annotated words must be re-calibrated due to the vocabulary problem. It also requires more time to determine and input the given keyword. On the contrary, browsing must work with one pre-defined keyword at a time and requires users to judge all possible pairs of images/keywords. But the effort to determine image relevance by browsing is usually much less than that by tagging, i.e., t_p, t_n is typically much smaller than t_f, t_s. Therefore, tagging is more suitable for annotating infrequent keywords, and browsing works better for frequent keywords.

5.3 Interactive Search

The complementary properties of tagging and browsing provide an opportunity to develop more efficient algorithms for manual image annotation by merging their strengths. Because our analysis suggests that tagging/browsing is suitable for infrequent/frequent keywords respectively, we develop an annotation algorithm by leveraging the powers of both annotation approaches in order to minimize overall annotation time across the entire corpus and the full set of annotation concepts/topics. We start with the automatic retrieval results and use machine learning techniques to assist in our annotation process. The entire annotation process lasted for 6 hours for 24 topics. More details of this system will be released in the final version of the notebook paper.

Overall, the only interactive run submitted by IBM achieves MAP of 0.36. A post-TREC analysis showed that applying simple temporal expansion at the end of the search result list, i.e., add discounted scores to the shots that are sufficiently close to the retrieved shots (within 3

shots) [Yan06], can boost the MAP to 0.43. The statistics of our annotation system shows that around 60% of the image-topic pairs in the entire video collection have been either browsed or tagged. Our system strikes a good balance on browsing and tagging the retrieved shots, where tagging produced 1529 retrieved shots and browsing produced 797. In contrast, according to our statistics, simple browsing interface can only annotate around 10% of the collection using the same amount of time. Simple tagging interface can annotate more image-topic pairs than browsing, however, it will miss most of the annotated shots provided by browsing.

6 Conclusion

IBM Research team participated in the TREC Video Retrieval Track Concept Detection and Search tasks. In this paper, we have presented preliminary results and experiments for both tasks. More details and performance analysis on all approaches will be provided at the TRECVID07 Workshop, and in the final notebook paper.

7 Acknowledgments

This material is based upon work supported by the US 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 US Government. We also thank Xu Han and Xun Xu for generating the HoG features.

References

[AQ07] S. Ayache and G. Quenot. Evaluation of active learning strategies for video indexing. In *Proceedings of Fifth International Workshop on Content-Based Multimedia Indexing (CBMI'07)*, 2007.

[Bre01] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[CCCP+04] J. C.-Carroll, K. Czuba, J. Prager, A. Ittycheriah, and S. B.-Goldensohn. IBM's PIQUANT II in TREC2004. In *NIST TREC Workshop*, 2004.

[CFPS02] D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 283–290. ACM Press, 2002.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 886–893, 2005.

[FLGD87] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Comm. of the ACM*, 30(11):964–971, 1987.

[fli] Flickr. `http://www.flickr.com`.

[HKM+99] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3), December 1999.

[HLY+06] A. G. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen. Extreme video retrieval: joint maximization of human and computer performance. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, 2006.

[HN06] A. Haubold and A. Natsev. Semantic multimedia retrieval using lexical query expansion and model-based reranking. In *International Conference on Multimedia and Expo(ICME)*, 2006.

[MMA+02] Y. Mass, M. Mandelbrod, E. Amitay, D. Carmel, Y. Maarek, and A. Soffer. JuruXML—an XML retrieval system. In *INEX '02*, Schloss Dagstuhl, Germany, Dec. 2002.

[NHT+07] A. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval: A comparative review and new approaches. In *ACM Multimedia (ACM MM)*, Sep. 2007.

[NNS04] A. Natsev, M. Naphade, and J. R. Smith. Semantic representation: Search and mining of multimedia content. In *ACM KDD*, 2004.

[NNT05] A. Natsev, M. R. Naphade, and J. Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM Multimedia*, Singapore, November 2005.

[NST+06] M. Naphade, J. R. Smith, J. Tešić, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. In *IEEE Multimedia Magazine*, volume 13, 2006.

[Pet] C. Petersohn. Fraunhofer HHI at TRECVID 2005: Shot boundary detection system. TREC Video Retrieval Evaluation Online Proceedings.

[TNS07] J. Tešić, A. Natsev, and J. R. Smith. Cluster-based data modeling for semantic video search. In *ACM International Conference on Image and Video Retrieval (ACM CIVR)*, 2007.

[TNXS07] J. Tešić, A. Natsev, L. Xie, and J. R. Smith. Data modeling strategies for imbalanced learning in visual search. In *International Conference on Multimedia and Expo(ICME)*, 2007.

[TTLW06] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1088–1099, 2006.

[vAD04] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2004.

[VSN05] T. Volkmer, J. R. Smith, and A. Natsev. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *Proceedings of the 13th ACM international conference on Multimedia*, 2005.

[XNT07] L. Xie, A. Natsev, and J. Tešić. Dynamic multimodal fusion in video search. In *International Conference on Multimedia and Expo(ICME)*, 2007.

[Yan06] R. Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2006.

[YNC07] R. Yan, A. Natsev, and M. Campbell. An efficient manual image annotation approach based on tagging and browsing. In *MS '07: Workshop on multimedia information retrieval on The many faces of multimedia semantics*, pages 13–20, 2007.

[YTS07] R. Yan, J. Tešić, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.