IBM Research TRECVID-2005 Video Retrieval System

Arnon Amir[*], Janne Argillander[†], Murray Campbell[‡], Alexander Haubold[‡],
Giridharan Iyengar[*], Shahram Ebadollahi[‡], Feng Kang[‡],
Milind R. Naphade[‡], Apostol (Paul) Natsev[‡], John R. Smith[‡], Jelena Tešić[‡], Timo Volkmer[‡]

Abstract

In this paper, we describe the IBM Research system for indexing, analysis, and retrieval of video as applied to the TREC-2005 video retrieval benchmark. We participated in the shot boundary detection, high-level feature extraction and search tasks and performed several new experiments in all the tasks. The paper describes the details of the approaches as well as the performance analysis. In general we observed good performance across all three tasks. In the detection task, we were able to achieve top mean average precision performance for all over 7 systems. In automatic search, we were able to achieve top mean average precision performance for 4 of our 6 automatic runs.

Keywords – Multimedia indexing, content-based retrieval, MPEG-7, LSCOM-lite, Support Vector Machines, Model vectors, Model-based reranking.

1 Introduction

We participated in the TREC Video Retrieval Track and submitted results for the following tasks:

1. **Shot boundary detection.** Our system is based on last year's system and uses a different MPEG decoder than the one which produced us color errors last year. Results improved very significantly, yielding high detection rates and highest gradual accuracy across all runs.

2. **Concept detection.** We focused on approaches that have been proven in the past to be successful while experimenting with fusion across features and approaches in a flat as well as hierarchical fashion. For all our 7 submissions we fused across multiple low-level features. For 6 of the 7 submitted runs we also fused across multiple approaches. We created internal partitions of the development data set and used various partitions, for training low-level feature-based models using individual approaches and for fusing across features and approaches. Unlike our past TRECVID submissions, we were unable to enforce cross-concept context [NS03a] due to time limitation. We choose our runs for submission objectively based on their performance on an internally held out set for selection purposes. Based on our previous experience across multiple TRECVID cycles [NSS04], we used support vector machines extensively for learning the mapping between low level features extracted from the visual modality as well as from transcripts and production related meta-features such as channel, language, time of the broadcast etc. We also built models for some if not all features extracted using three other approaches: a modified nearest neighbor learner, a maximum entropy learner and a Gaussian mixture model. For some high-level features from the benchmark which had enough training samples, and were predominantly regional, we also applied an extension of a new generalized multiple instance learning algorithm [NS05]. For fusing across approaches and features, we resorted to simple fusion techniques. In one approach we fused across features and approaches at the same time. In the other we first fused across features and then across approaches. We tried fusing with simple normalization and ensemble averaging, with weights learning for a limited number of approaches and with validity weighting [SNN03]. Based on all the experiments we submitted the following 7 runs:

   - A_JW_ABOF1: Best of Fusion across features and approaches selected individually for each concept
   - A_JW_A1SA2: Flat fusion across all features and approaches using statistical normalization and ensemble averaging
   - A_JW_A1SV3: Flat fusion across all features and approaches using statistical normalization and validity weighting
   - A_JW_ABOA4: Hierarchical fusion across features for each approach and validity weighted fusion across all approaches
   - A_JW_SVM5: Fusion across all features for each approach and selection of the best single approach for each concept,

[*]IBM Almaden Research Center, Palo Alto, CA, USA
[†]IBM Global Services, Helsinki, Finland
[‡]IBM T. J. Watson Research Center, Hawthorne, NY, USA

- A_JW_M2SW6: Fusion across all features and 2 approaches SVM and MECBR with weights learning
- A_JW_SVMFD7: SVM models with fusion across all approaches trained on the entire development set with optimal parameters from one partition

Results indicate that two approaches A_JW_SVMFD7 and A_JW_ABOA4 topped performance with a mean average precision of 0.3356 whereas the fusion across a subset of approaches A_JW_M2SW6 returned the lowest MAP of 0.3126 across the 7 runs. All the 7 runs resulted in the top 7 MAPs across all the TRECVID runs evaluated.

3. **Search.** We focused heavily on automatic search this year, building fully automatic retrieval systems for both speech and visual modalities and producing the top runs among automatic type A search systems. We used a new text search engine for our speech-based retrieval system and explored three automatic query refinement methods for it. They were fused to generate a text-only baseline (run F_A_1_JW_T_7) of 0.057 MAP. For our visual retrieval system, we applied a combination hypothesis of two complementary light-weight learning approaches—SVM and MECBR—which for the first time significantly outperformed speech-based retrieval (run F_A_2_JW_V_3 with MAP of 0.110), likely due to the inconsistent quality of ASR+MT transcripts on foreign broadcast videos. Finally, we developed methods for model-based re-ranking of both retrieval hypotheses based on SVM models we built for the 39 concepts annotated by TREC participants. Model-based re-ranking improved our text-only baseline to a MAP of 0.070 (run F_A_2_JW_TM_6) and our visual-only baseline to a MAP of 0.119 (run F_A_2_JW_VM_4). This year we used simple query-independent non-weighted fusion methods for combining our speech-based and visual runs, and while we did not observe gains from fusing the two modalities, our parameter-free fusion approach was able to generalize fairly well, considering the wide performance gap we observed between the two modalities. Our text+visual multimodal run (with ID F_A_2_JW_TV_5) essentially matched the performance of the better modality producing a MAP of 0.106. Similarly, our model re-ranked text+visual multimodal run (with ID F_A_2_JW_TVM_2) matched the performance of the re-ranked visual-only run, generating a MAP of 0.119. Overall, our new visual retrieval approach and the model-based re-ranking approach were the most significant performance contributors for our system.

In the paper, we describe the IBM Research system and examine the approaches and results for each run. The video content is analyzed in an off-line process that involves automatic shot boundary detection, 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.

2 Shot detection

IBM has submitted ten runs to the Shot Boundary Detection task. All runs use the same algorithm with minor modifications. The main change from our last year system is not in the algorithm but in replacing the MPEG-1 decoder.

In TRECVID 2004 the video encoding of most of the TRECVID videos was encoded using an IPPPPPPPPPPPPP GOP (Group of Pictures) structure, instead of the more common IPBBPBBPBBPBBPB sequence. Interestingly, decoding this GOP with the commercial decoder we used introduced cumulative color errors while decoding the video frames through each GOP, growing up to 3 percents RMS at the last P-frame of the GOP and dropping to zero at the next I-frame. Those errors are hardly noticeable in regular video playback. however, when processed by the SBD algorithm, those color errors cause increased distances between color histograms of pairs of frames. This in turn causes an increase in all the adaptive thresholds, computed using statistics of frame differences over a symmetric window of 61 frames centered around the processed frame and used throughout the system [AHI+03]. The higher thresholds, combined with the perceived frame noise, caused a significant decrease in detection recall and an overall degraded performance. It was the first and only time we observed this type of decoder color error, and only with this type of GOP encoding. Replacing the decoder with a different one after TRECVID2004 resolved this problem.

We used the test of 2004 as our training set for 2005. Table 1 shows processing results of test set of TRECVID04. Runs are sorted by decreasing F# of the Detection of All Changes criteria. The ten runs with system ID-s of TRECVID05 are clearly much better than the other five runs, four of which representing our best submitted SBD system in a different year, from TRECVID01 to TRECVID04, and the fifth one, marked N171QT was obtained by converting all the MPEG-1 videos of the test set to MPEG-4 using a QuickTime player, and processing the MPEG-4 files. This last run was a test that shows a very significant improvement in gradual recall over the N171 run, using the same executable and transcoded video sequences. Moreover, it showed no ”noise” in the processing, which led us to look more carefully at the differences between decoders. As said, the single main cause for improvement

Table 1: Shot boundary detection results, run and evaluated with TRECVID 2004 test set. Labels in the leftmost column correspond to the TRECVID year in which the system was submitted.

TRECVID	SysID	AllF#	AllRcl	AllPrc	CutsF#	CutsRcl	CutsPrc	GradF#	GradRcl	GradPrc	GrAcF#	GrAcRcl	GrAcPrc
TVID05	N209	0.892	0.872	0.912	0.933	0.935	0.931	0.798	0.741	0.865	0.881	0.846	0.918
TVID05	N208	0.890	0.875	0.906	0.929	0.931	0.928	0.803	0.758	0.854	0.876	0.842	0.913
TVID05	N207	0.890	0.873	0.907	0.928	0.932	0.925	0.802	0.750	0.862	0.878	0.841	0.919
TVID05	N212	0.890	0.874	0.906	0.928	0.933	0.924	0.802	0.750	0.862	0.877	0.840	0.918
TVID05	N210	0.889	0.873	0.906	0.929	0.932	0.927	0.799	0.750	0.856	0.880	0.846	0.917
TVID05	N204	0.889	0.871	0.909	0.928	0.929	0.928	0.801	0.749	0.861	0.867	0.843	0.892
TVID05	N202	0.883	0.883	0.882	0.915	0.939	0.892	0.809	0.766	0.857	0.874	0.827	0.927
TVID05	N198	0.882	0.878	0.885	0.918	0.940	0.898	0.797	0.748	0.854	0.878	0.833	0.929
TVID05	N197	0.881	0.878	0.885	0.918	0.939	0.898	0.797	0.749	0.853	0.878	0.832	0.929
TVID05	N194	0.881	0.886	0.876	0.916	0.940	0.894	0.802	0.774	0.833	0.877	0.841	0.915
POST04	N171QT	0.855	0.825	0.888	0.892	0.885	0.900	0.769	0.698	0.857	0.864	0.836	0.894
TVID02	N047	0.844	0.822	0.867	0.891	0.915	0.867	0.726	0.625	0.866	0.744	0.646	0.879
TVID04	N171	0.832	0.774	0.898	0.897	0.899	0.896	0.655	0.512	0.909	0.845	0.804	0.890
TVID01	Nalm1	0.774	0.724	0.831	0.863	0.903	0.827	0.494	0.347	0.855	0.640	0.480	0.961
TVID03	N127	0.765	0.685	0.866	0.861	0.862	0.861	0.463	0.311	0.902	0.687	0.534	0.965

Table 2: Shot boundary detection results, run and evaluated with TRECVID 2005 test set. The official submission runs of this year are marked with TVID05. run are sorted in decreasing F# value of detection of All Boundaries.

TRECVID	SysID	AllF#	AllRcl	AllPrc	CutsF#	CutsRcl	CutsPrc	GradF#	GradRcl	GradPrc	GrAcF#	GrAcRcl	GrAcPrc
MAX		0.897	0.894	0.901	0.942	0.936	0.949	0.789	0.788	0.791	0.852	0.833	0.871
TVID05	N204	0.876	0.909	0.845	0.912	0.933	0.891	0.776	0.838	0.722	0.836	0.836	0.837
TVID05	N212	0.875	0.914	0.839	0.911	0.936	0.887	0.775	0.848	0.714	0.850	0.827	0.875
TVID05	N207	0.875	0.912	0.840	0.912	0.936	0.890	0.772	0.842	0.712	0.851	0.827	0.877
TVID05	N208	0.871	0.913	0.832	0.910	0.936	0.886	0.764	0.848	0.695	0.852	0.833	0.871
TVID05	N209	0.869	0.916	0.826	0.906	0.940	0.875	0.765	0.843	0.700	0.847	0.824	0.872
TVID02	N47	0.867	0.891	0.845	0.899	0.935	0.867	0.770	0.764	0.776	0.820	0.748	0.908
TVID05	N210	0.863	0.915	0.817	0.902	0.937	0.870	0.759	0.853	0.683	0.843	0.818	0.869
TVID01	Nalm1	0.854	0.903	0.811	0.885	0.937	0.839	0.762	0.803	0.726	0.814	0.729	0.923
TVID05	N202	0.854	0.912	0.803	0.888	0.935	0.846	0.760	0.848	0.688	0.850	0.824	0.878
TVID05	N198	0.849	0.917	0.791	0.882	0.940	0.831	0.758	0.850	0.684	0.844	0.812	0.879
TVID05	N197	0.849	0.918	0.790	0.882	0.940	0.830	0.757	0.851	0.682	0.844	0.812	0.879
TVID03	N127	0.843	0.865	0.822	0.877	0.895	0.860	0.746	0.778	0.718	0.836	0.765	0.922
TVID05	N194	0.841	0.919	0.776	0.883	0.938	0.834	0.733	0.864	0.637	0.846	0.823	0.870
TVID04	N171	0.839	0.869	0.812	0.876	0.905	0.849	0.732	0.764	0.703	0.793	0.806	0.780

with the 2005 systems on this data set was the change of decoder.

Table 2 shows the processing results of the same systems on the test set of TRECVID05. Compared to Table 1, most of the ten 2005 systems maintain relatively similar performance between the two test sets, TRECVID04 and TRECVID05. System N204 top our 2005 submissions and is different in that in computing the color histograms it ignores the bottom 20 percents of the frame, which often contains overlay text. When excluded from the histogram, it slightly improves the discrimination between frames across different shots. The row marked as MAX contains in each of the four evaluation criteria the values obtained by the best of all 163 runs in TRECVID05, when ranked by the F# of that criteria. These maximum results correspond to three different runs, marked as hu26, bs-8 and N208. Our best detection rates are delivered by system N204, only slightly below the best performing run this year. System N208, a hair away in detection rates from N204, produced the best F# for Gradual Accuracy (0.876) across all 163 submitted runs. Our past systems performed better on the 2005 data set than on the 2004 data set, despite of using the old decoder. Evidently the encoding of the test set this year is of the more common GOP sequence, as oppose to last year.

3 Video Descriptors

3.1 Visual Features

The system extracts eight different visual descriptors at various granularities for each representative keyframe of the video shots. Relative importance of one feature modality vs. another may change from one concept/topic to the next, the relative performance of the specific features within a given feature modality (e.g., color histogram vs color correlogram) should be the same across all concepts/topics, and can therefore be optimized globally for all concepts and topics.

The goal of feature selection was to optimize globally the feature type and granularity for each feature modality so that the fusion across modalities gives optimal results both for concept detection and search task. We performed extensive experiments using the TRECVID 2005 development set and TRECVID 2003 query topic to select the best feature type and granularity for color and texture modalities for concept detection and search tasks, respectively.

The following descriptors had the top 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 Layout (EHL)—localized edge histograms with 8 edge direction bins and 8 edge magnitude bins, based on a Sobel filter, extracted from a 5-region layout consisting of four corner regions and a center overlapping region (320-dimensional).

Although, the described visual descriptors are very similar to the MPEG-7 visual descriptors [MSS02], they differ in a sense that they have been primarily optimized for retrieval and concept modeling purposes, with much less consideration given to compactness or computational efficiency.

3.2 Motion Features

We introduce a novel low-level visual feature that summarizes motion in a shot. This feature leverages motion vectors from MPEG-encoded video, and aggregates local motion vectors over time in a matrix, which we refer to as a motion image. The resulting motion image is representative of the overall motion in a video shot, having compressed the temporal dimension while preserving spatial ordering.

Motion vectors are present for all macroblocks in P and B frames of MPEG video. For I-frames, which start a GOP sequence of P and B frames, motion vectors have zero-magnitude. We generate a new image for each shot with dimensions equal to the matrix of macroblocks. For TREC news videos, motion images are dimensioned 20 columns by 13 rows. We preserve the spatial location of macroblock motion vectors by placing the vector's origin in the corresponding position in the motion image. We scale each vector by some constant factor F, which represents the predicted future direction of that vector over F-many frames. The scaled vector is added to the motion image, which aggregates all such vectors for the entire shot. The resulting two-dimensional motion image is cropped, linearized, and normalized, and used as a feature vector. In the case of TREC videos, this vector contains 260 features, corresponding to a scanline-version of the motion image.

3.3 Text Features

We extracted several text features for each shot based on the speech transcript corresponding to the shot after expansion of the shot boundaries to include up to 5 immediate neighbors on either side without crossing full video clip boundaries. This shot expansion results in overlapping speech segments and attempts to compensate for speech and visual mis-alignment. The resulting shot documents were then processed for stop-word removal and Porter stemming, and for each term, the following text features were computed:

- Term Frequency (TF) in given shot document
- Inverse Document Frequency (IDF) across all shot documents
- TF*IDF
- Binary term flag, 0 or 1, indicating presence or absence of given term in given shot document

Each shot was then represented in a sparse vector format, where the ith dimension reflected one of the above measures for the ith term in the speech vocabulary. These features were used for SVM-based modeling in the High-Level Feature Extraction task.

3.4 Semantic Features

The third feature modality we used was that of 39-dimensional semantic model vectors built from the detection confidence scores with respect to 39 LSCOM-lite concepts. Extraction of the model vector features based on the semantic modeling is described in detail in section 4.

4 Concept Modeling

Our basic principle for modeling semantic concepts or high-level features based on low-level media features has consistently been to apply a learning algorithm to the low-level features [NBS+02, NJ03, NLN+03, NSS04, NNT05a]. Our 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. Figure 1 illustrates our concept detection pipeline.

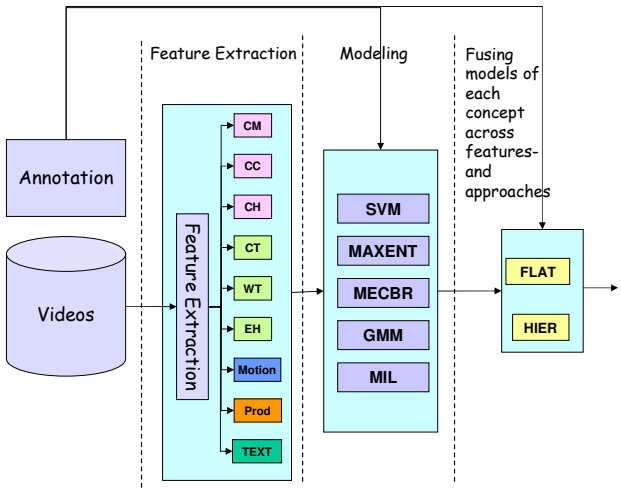

Figure 1: The IBM TRECVID 2005 Concept Detection pipeline.

4.1 Data Partitioning and Parameter Optimization

We partitioned the development data set provided by NIST into the following 4 internal partitions for facilitating hierarchical processing experiments and selection by randomly assigning videos from the development set to each partition. The list below gives approximate number of keyframes in each partition.

- Training Set: 41K keyframes
- Validation Set: 7K keyframes
- Fusion Set: 7K keyframes
- Selection Set: 7K keyframes

4.2 Learning Approaches

This year we focused on approaches that have been proven in the past to be successful while experimenting with new techniques for cross granularity propagation. We then proceeded to fuse across features and approaches in a flat as well as hierarchical fashion. For all our 7 submissions we fused across multiple low-level features. For 6 of the 7 submitted runs we also fused across multiple approaches. We created internal partitions of the development data set and used various partitions, for training low-level feature-based models using individual approaches and for fusing across features and approaches. We used support vector machines extensively for learning the mapping between low level features extracted from the visual modality as well as from transcripts and production related meta-features such as channel, language, time of the broadcast etc. We also built models for some if not all features extracted using three other approaches: a modified nearest neighbor learner, a maximum entropy learner and a Gaussian mixture model. For some high-level features from the benchmark which had enough training samples, and were predominantly regional, we also applied an extension of a new generalized multiple instance learning algorithm [NS05].

4.2.1 Support Vector Machines

We represent keyframes with a set of low-level visual features, such as colors, textures, and shapes, and motion. We also extract a bunch of production meta-features. For the visual features and production meta-features we use support vector machines with non-linear kernels. We also represent shots by text features extracted from the transcripts and then convert them to sparse representation formats. We use support vector machines with linear kernels for the text features.

In the training phase, we learn feature representations corresponding to the binary hypotheses for each concept (presence/absence) using support vector machines [Vap95].

Support Vector Machines are popularly used for classification and regression in various domains including the multimedia domains. For the past few years support vector machine classifiers have resulted in top performance in concept detection for NIST TRECVID evaluations [NJ03, NLN$^+$03, NSS04, NNT05a]. Support vector machines used with nonlinear kernels allow us to learn nonlinear decision boundaries even when the data is high dimensional and are not affected by the curse of dimensionality due to the way the optimization is formulated to minimize empirical risk. They also offer good generalization capability. For the concept detection experiments there has been extensive reporting of the use of support vector machine classifiers and the procedures for tuning the model parameters including kernel parameters.

We use the held out validation set in selecting model parameters as well selecting optimally performing features from across all low-level features. We use the Radial Basis Kernel for the SVM experiments.

Performance of SVM classifiers can vary significantly with variation in parameters of the models. Choice of the kernels and their parameters is therefore crucial. To minimize sensitivity to these design choices, we experiment with different kernel parameters. Radial basis function kernels usually perform better than other kernels. In our experiments we build models for different values of the RBF parameter γ (variance), relative significance of positive vs. negative examples j (necessitated also by the imbalance in the number of positive vs. negative training samples) and trade-off between training error and margin c. While a coarse to fine search is ideal, we try several values of γ, j and c thus evaluating dozens of configurations. Using the validation set we then performed a grid search for the

combination that resulted in highest performance measure value, where this measure is the non-interpolated average precision over 1000 retrieved shots as a measure of retrieval effectiveness. Let R be the number of true relevant documents in a set of size S; L the ranked list of documents returned. At any given index j let R_j be the number of relevant documents in the top j documents. Let $I_j = 1$ if the j^{th} document is relevant and 0 otherwise. Assuming $R < S$, the non-interpolated average precision **(AP)** is then defined as

$$\frac{1}{R}\sum_{j=1}^{S}\frac{R_j}{j} * I_j \quad (1)$$

In the detection phase we use the optimal models to evaluate the target images for the presence/absence of the concept and generate a confidence measure correspondingly that can then be used to rank images for each concept.

4.2.2 Gaussian Mixture Models

We built gaussian mixture models for all the benchmark concepts using mixtures of diagonal Gaussians. This approach is known to work for concepts with a large number of training samples but results in lower performance than SVM models for concepts with small number of training samples [vtr02, NJ03]. We build conditional density models for positive samples and negative samples and then used the likelihood ratio test to generate the ranking at detection time.

4.2.3 Maximum Entropy Methods

In MaxEnt modeling, we assume that a random process produces an output (label) y given a context x. In multimedia annotation, y, which is a member of a finite set (vocabulary) Y, can be seen as a label for a specific shot. And x, a member of a finite set X, as extracted information (features) from the current frame. Training data is presented in pairs $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. The task is to learn possible correlations between x and y, and to build statistical models that can be used to annotate previously unseen shots automatically. The empirical probability distribution function (pdf) based on training data is as follows

$$\tilde{p}(x, y) = \frac{1}{n} freq(x, y) \quad (2)$$

Where *freq* is the count of a specific pair (x, y) in the training data. In real world applications, the training set size is finite. Therefore, the empirical distribution is a poor estimate of the joint pdf. Based on this partial information, MaxEnt modeling can be used to estimate the pdf that generated the empirical distribution $\tilde{p}(x, y)$ in an unbiased way[Jay57]. At the core of the modeling process are *predicates*. These predicates are used to specify constraints on the model. In MaxEnt, the process of defining predicates is central to modeling: The goodness of the models is dependent on the ability of these predicates to capture relevant information and we differentiate from previous work in this [JM04].

In our experiments, we extract 3 types of low-level image features from each video shot: Lab space color moments, edge histograms and summary statistics of grey-level co-occurrence matrices. Together, these form our 3 different low-level descriptors which we will term *Color*, *Edge* and *Texture* in further discussions. Furthermore, we partition each shot key-frame (comprising 350×240 pixels) into 35 regions (50×48 pixels each) and extract the feature descriptors for each of these 35 regions.

We try several predicates including unigram predicates, location dependent unigram predicates, and 2 types of bigram predicates. Both types of bigrams are constructed by combining the tokenized features in the product space of the unigram predicates. This choice imposes the possibility of obtaining bigram values that are not supported in the training data, resulting primarily from the sparseness of the product space. To counter this, we employ an approach inspired from class-based language models in speech processing. When two unigrams are composed into a bigram, we treat them differently. We start with few clusters for the composed unigrams and slowly increase the number of clusters such that the number of unique bigram predicates observed (in the training data) at each step matches the total possible bigram product space values. We stop at the largest cluster size for which this condition is met in the training data. The above predicates model individual low-level feature descriptors (i.e. Color, Edge, Texture). We then develop predicates predicates that model the interactions between the various low-level feature descriptors where the joint observation predicate is active only if all low-level descriptors are present in a given region. For more precise description please see [mst] and [J. 05].

4.2.4 Modified Nearest Neighbors—MECBR

Multi-example content-based retrieval (MECBR) is a modified nearest neighbor classifier used typically in content-based retrieval (CBR) settings [NS03b, NNT05b]. Previously, we used MECBR to generate fully automatic visual runs for the TRECVID 2003 and 2004 search tasks [AHI$^+$03, ACF$^+$04]. This year, in addition to the search task, we explored MECBR for the concept detection task as well. MECBR models a query topic or a concept from a set of positive examples by dividing the examples into visually distinct categories, selecting representatives from each category, and treating each representative example as an independent CBR query. The results from the multiple queries are then aggregated within and across categories into a single ranked list. Parameters of

the method determine how exactly the individual result lists are fused together, with alternatives ranging from simple non-weighted fusion (e.g., OR fusion logic across examples) to complex weighted Boolean fusion (i.e., a mixture of AND/OR fusion with weights) [NS03b]. The end result can therefore be thought of as the fusion of several distance-weighted nearest neighbor classifiers, each working with a different subset of the positive examples. This allows for different score normalization and aggregation methods in different portions of the feature space, which is an advantage over the traditional k-nearest neighbor formulation. The disadvantage of this method is in the computational overhead of executing a separate CBR query potentially for each positive example. For search scenarios where the number of positive examples is very limited (e.g., less than 10), this is not a significant overhead. However, in the case of concept modeling, where the training set may contain thousands of positive examples, this approach is clearly not scalable. To reduce the computational requirements of the approach for modeling of frequent concepts, we sample the positive examples using a biased sampling method which iteratively selects the most visually distinct examples from a given set. This is based on the idea of anchoring, where each successive anchor is selected so that it is as far away as possible from the previously selected anchors [NS03c]. For concept detection, we sampled up to 800 distinct positive examples for each concept and treated them as independent category representatives.

The above sampling approach naturally tends to pick outliers, however, so it is very sensitive to noise in the training set, where some examples may be mislabeled. To mitigate this, we used annotation redundancy, where available, to associate a relevance score to each shot. In particular, when deciding how to label a shot with redundant but conflicting annotations, we considered three different policies for aggregating the overlapping annotations. The first policy was a liberal one where a shot was labeled positive if any of its annotations were positive. This was most applicable to very rare concepts, where a false negative can be more detrimental than a false positive in the annotation. The second policy was a strict one, requiring a perfect agreement across all annotators in order to mark a shot positive. This policy was most conservative, resulting in the smallest set of positive examples, and was therefore applicable only to the most frequent concepts where false positives can be much more damaging than false negatives. The third policy was a 2/3 majority vote-based annotation and was most appropriate for concepts that were neither too rare nor too frequent. The optimal annotation resolution policy as well as the optimal score normalization and fusion parameters for each concept-feature combination were then determined based on a held-out validation set performance.

4.2.5 Multiple Instance Learning

Statistical learning techniques provide a robust framework for learning representations of semantic concepts from multimedia features. The bottleneck is the number of training samples needed to construct robust models. This is particularly expensive when the annotation needs to happen at finer granularity. It is precisely due to the cost of regional annotation, that the TRECVID 2005 Common Annotation exercise only involved frame-level annotation. We experiment with a novel approach where the annotations may be entered at coarser spatial granularity while the concept may still be learnt at finer granularity. Using the multiple instance learning paradigm, we learn representations of concepts occurring at the regional level by using annotations for several images. We use an extension of the generalized multiple instance learning algorithm [NS05] that can scale to a large number of training samples as well as a large number of instances per bag. The algorithm also provides the ability to plug in different density modeling or regression techniques.

The essence of applying multiple instance learning to disambiguate across granularity is shown in Figure 2. Here we use the same notation of *Bags* and *Instances* as in [MLP98]. A Bag is a collection of instances. Annotation is provided at the bag level but actually reflects the label of one or more instances in that bag. If at least one instance (region) that is positive the corresponding bag is labeled positive. Conversely a bag is labeled negative when all instances (regions) are negative for the semantic concept. The problem is to then learn in some feature space a concept point or a set of concept points that are closest to maximum possible positive bags (i.e. instances in these bags) and simultaneously away from as many negative bags (i.e. negative instances) as possible. Figure 2 uses a 2 dimensional feature space to illustrate this idea.

For TRECVID we applied multiple instance learning only to a limited number of concepts for which annotation was sufficient and the criteria to leverage multiple instance learning applied.

Figure 3 compares the 4 approaches, applied to all benchmark concepts using an internal partition (Fusion set)

4.3 Fusion

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 optimal fusion configuration based on a held-out validation set performance. Fusion parameters include a score normalization method and a score aggregation method. Score normalization methods include range normalization, statistical normalization shifting the score distribution to zero mean and

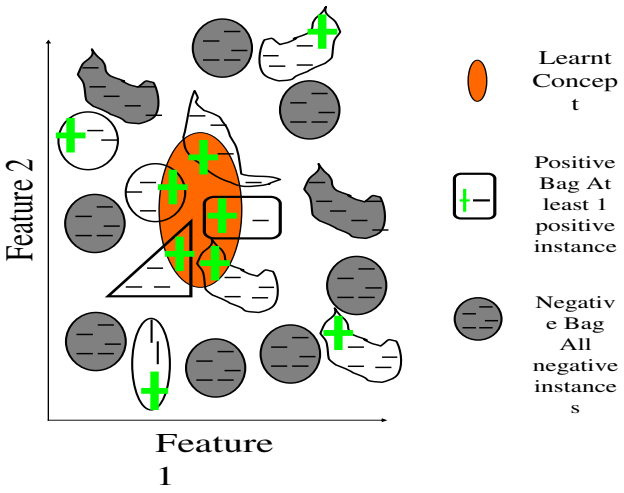

Figure 2: This figure shows a distribution of *bags* in a two dimensional feature space. Only bags are labeled. Multiple instance learning can result in the region in orange as the target concept as it is closest to as many positive bags as possible while farthest from many negative bags.

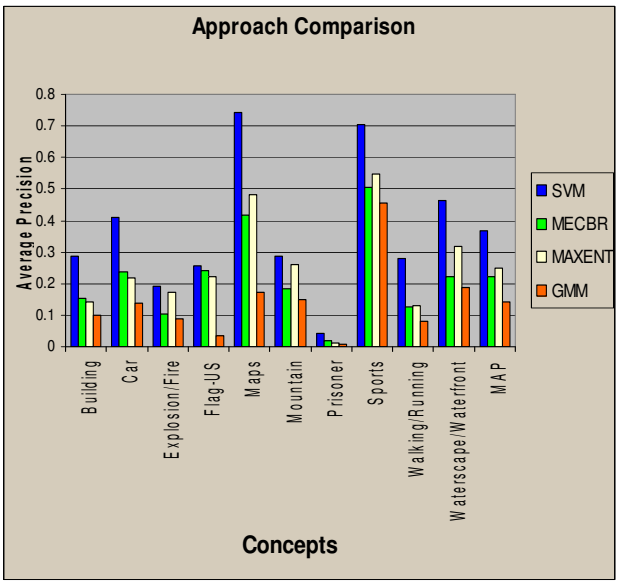

Figure 3: This figure compares the performance of the 4 approaches across the 10 benchmark concepts on the Fusion Set. SVMs outperform all other approaches by a wide margin.

uni-variance, Gaussian normalization, and rank normalization which discards the absolute scores and uses only the rank of each item in the result list. The fusion methods we considered include MIN, MAX, AVG, and weighted AVG 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 explored two main fusion variations depending on the order in which we fused hypotheses.

Flat Fusion across Features and Approaches. The first approach was based on a single-level global fusion across all individual hypotheses, regardless of whether they came from different features or modeling techniques. We call this *flat fusion*. With this approach we performed a full grid search in the fusion parameter space but due to the large number of hypotheses being fused, we explored only binary weights (presence or absence of each hypothesis) with the weighted average score aggregation method. This has the effect of doing hypotheses selection but only non-weighted fusion.

Hierarchical Fusion across Approaches. The other approach was based on hierarchical, two-level fusion, where all features were fused first for each modeling approach, followed by fusion across the independent modeling approaches. This *hierarchical fusion* limits the number of hypotheses being fused at the second level and significantly reduces the fusion parameter search space. We were therefore able to explore more weighted combinations at this level by considering 10 uniformly distributed weight values for each dimension.

4.4 Building Model Vector

We built SVM models for all 39 concepts of the LSCOM-lite lexicon [NKK+] shown in Figure 4.

For each concept we applied the same training and validation procedure as applied to the ten benchmark concepts. The only exception was that in the interest of time we were able to fuse across only 4 features i.e. color correlgram, color moments on a 3×3 grid, global cooccurrence texure and wavelet texture on a 3×3 grid. We fused across these features using statistical normalization and ensemble averaging. The performance of the 39 concepts on an internal partition, is shown below in Figure 5.

Once these 39 concepts are detected, we then stack them together to create model vectors. These vectors are used along with low-level features in our visual search experiments. The models are also used for reranking in our search experiments describes in Section 5

4.5 Experiments and Results

Based on all the experiments we submitted the following 7 runs:

- A_JW_ABOF1: Best of Fusion across features and approaches selected individually for each concept

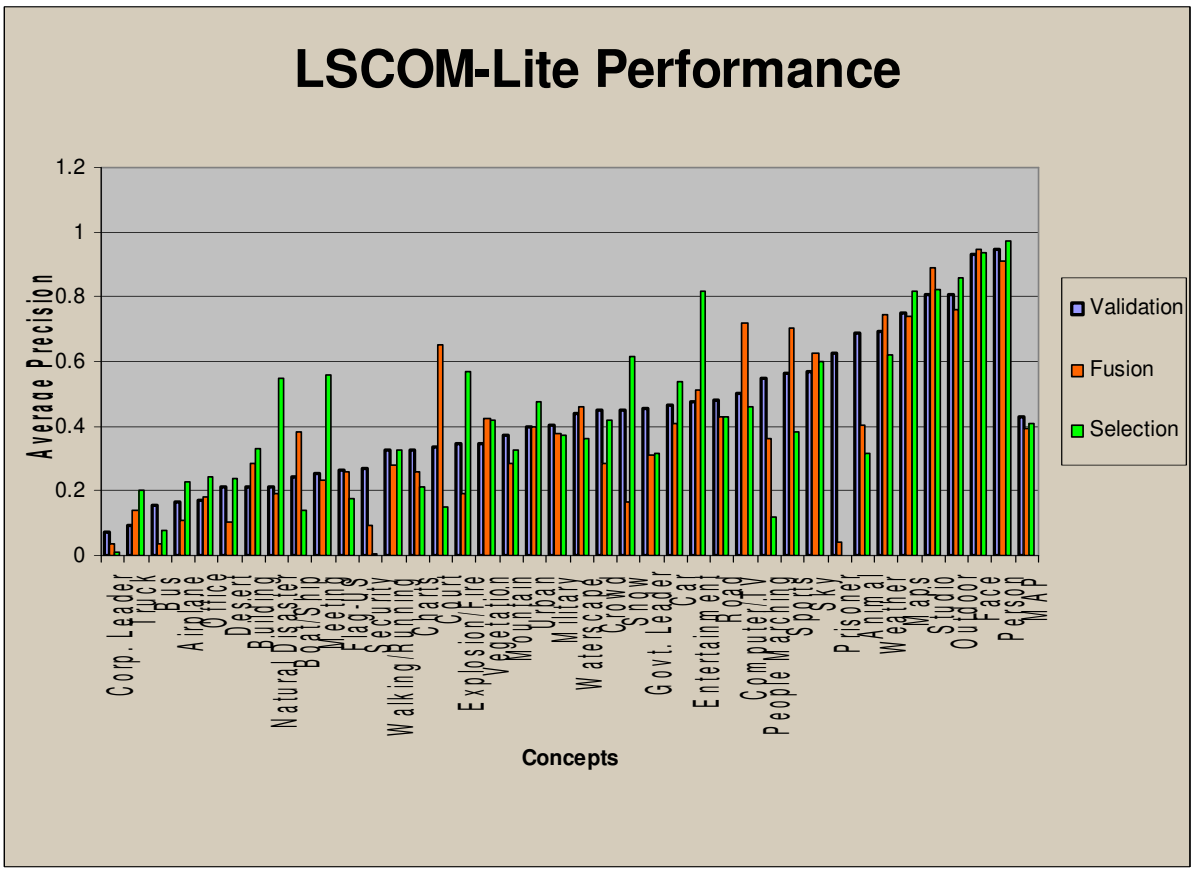

Figure 5: The performance of the modeled LSCOM-lite lexicon on an internal test set [NKK+].

- A_JW_A1SA2: Flat fusion across all features and approaches using statistical normalization and ensemble averaging
- A_JW_A1SV3: Flat fusion across all features and approaches using statistical normalization and validity weighting
- A_JW_ABOA4: Hierarchical fusion across features for each approach and validity weighted fusion across all approaches
- A_JW_SVM5: Fusion across all features for each approach and selection of the best single approach for each concept,
- A_JW_M2SW6: Fusion across all features and 2 approaches SVM and MECBR with weights learning
- A_JW_SVMFD7: SVM models with fusion across all approaches trained on the entire development set with optimal parameters from one partition

Results indicate that two approaches A_JW_SVMFD7 and A_JW_ABOA4 topped performance with a mean average precision of 0.3356 whereas the fusion across a subset of approaches A_JW_M2SW6 returned the lowest MAP of 0.3126 across the 7 runs. All the 7 runs resulted in the top 7 MAPs across all the TRECVID runs evaluated.

Figure 6 shows the IBM runs in comparison to the rest of the submitted runs for TRECVID. The blue colored bars denote the 7 IBM runs and all result in top MAP performance.

A concept-wise comparison with the mean, median and best non-IBM run is shown in Figure 7

5 Search

5.1 Automatic Search

The IBM team focused heavily on automatic search for this year's TRECVID, submitting 6 automatic runs out of 7 allowed submissions. Our automatic search system was a combination of speech-based retrieval with automatic query refinement, visual retrieval using a combination of two light-weight learning approaches, and model-based reranking using automatic concept detectors for the 39 concepts from the TRECVID 2005 common annotation effort. All processing was done at the sub-shot level based on the master shot boundary reference provided by the Fraunhofer

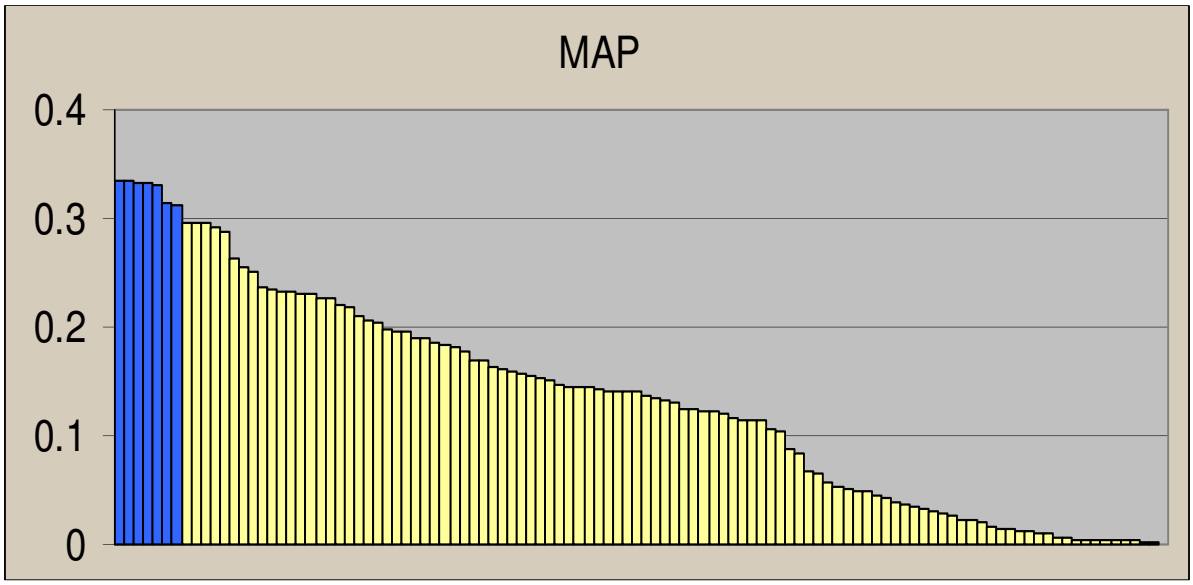

Figure 6: IBM Concept Detection runs compared to all TRECVID 2005 concept detection system runs. IBM runs represented by blue bars. return MAP between 0.3126 and 0.3356.

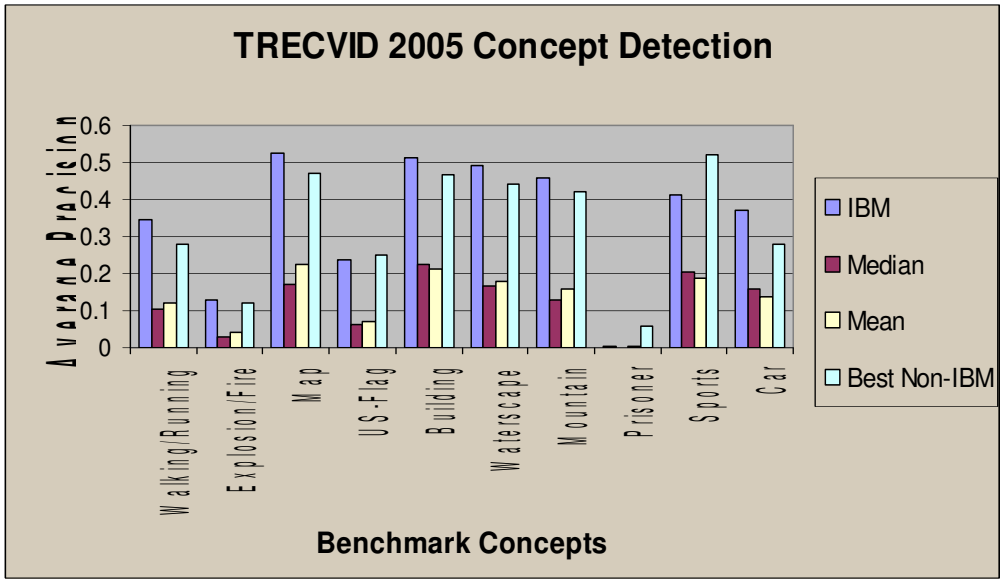

Figure 7: IBM runs result in top performance for 7 of the 10 benchmark concepts with performance above the median in the rest.

Institute [Pet], where each sub-shot was represented by a single keyframe and a corresponding speech transcript segment. Results were then aggregated at the shot level by taking the maximum confidence score across all sub-shots for each shot. The overall system is illustrated at a high level in Figure 8.

5.1.1 Speech-based retrieval

Our speech-based retrieval system this year was built using the IBM *Unstructured Information Management Architecture (UIMA)* [FL04] and the JuruXML semantic search engine [MMA+02] developed by IBM Research. The UIMA SDK is scheduled to be open-sourced by the end of the year and is currently available for download at the IBM Alphaworks site [uim]. The SDK also includes the JuruXML semantic search engine. In addition to the base UIMA SDK,

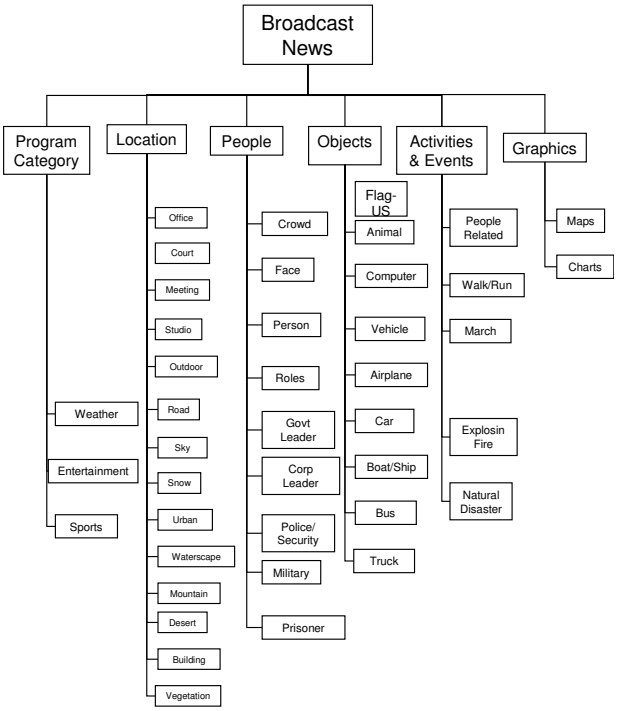

Figure 4: The modeled LSCOM-lite lexicon [NKK+]

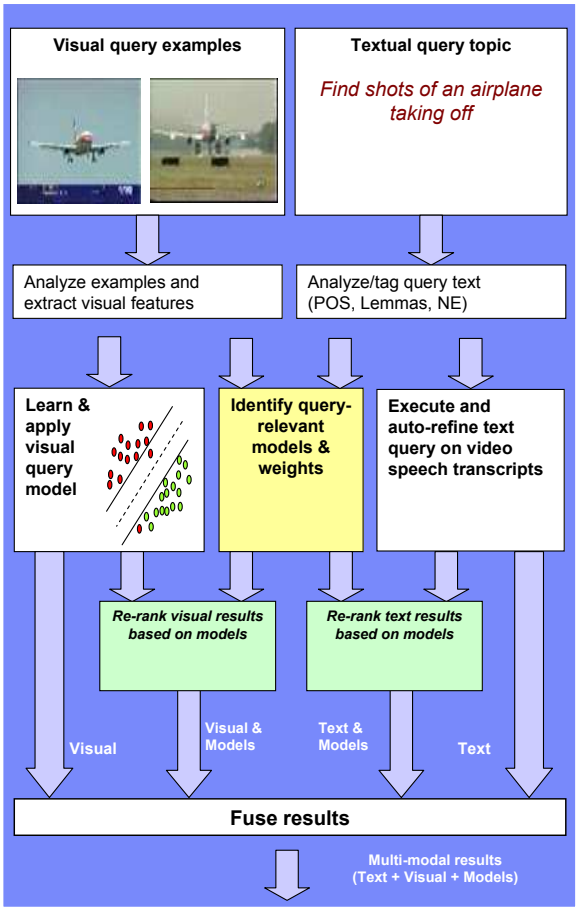

Figure 8: Overview of IBM automatic search system.

we used several UIMA components developed by IBM Research for advanced text analytics. These include the TALENT system for Text Analysis and Language Engineering Technology, the Resporator (RESPOnse geneRATOR) system [PBCR00] built on top of TALENT, and the PIQUANT Question Answering system [CCCP+04] built on top of RESPORATOR. We used the TALENT component to perform token and sentence detection, lemmatization, and part-of-speech annotation. The RESPORATOR component was used to annotate text with over 100 semantic categories, including both named and unnamed entities, such as people, roles, objects, places, events, etc. It is a rule-based annotator developed originally for Question Answering purposes [PBCR00] and used extensively by the PIQUANT system. Finally, we leveraged the query analysis and refinement capabilities of PIQUANT in order to do automatic query expansion to the categories detected by RESPORATOR. For example, a query containing the term "basketball" would automatically be expanded to include the "SPORTS" tag detected by the RESPORATOR component. This essentially performs automatic query sense disambiguation and expansion.

In addition to the RESPORATOR-based query expansion, we explored two other methods for automatic query refinement based on pseudo-relevance feedback [XC96], which are based on the assumption that the top-ranked documents for a given query are indeed relevant. Traditional relevance feedback methods such as Rocchio refinement process [Roc71] can then be used to effectively refine the query. In particular, a set of top-ranked documents is first retrieved using the original user query. The weight of the query terms is modified according to their frequency in this set. In addition, expansion terms are selected from this set, based on various selection criteria, and added to the query. The refined query is then submitted to the system, resulting in the final set of documents considered relevant to the original user query. An alternative way to select additional terms for query expansion is to consider *lexical affinities (LA)*, which are pairs of terms that frequently co-occur within a close proximity of each other (e.g., phrases). The idea is that if one of the terms in a lexical affinity appears in the query text, it is likely that the other part of the LA is also relevant. An LA-based query expansion method was proposed in [CFPS02]. We used both automatic query expansion approaches since both are available as native functionality in the JuruXML search engine. Our final speech-based retrieval system was therefore the combination of three separate automatic query refinement methods—QA-based query expansion to text categories, Rocchio-based pseudo-relevance feedback query expansion, and lexical

affinity-based pseudo-relevance feedback query expansion. The parameters for each of the methods were tuned globally on the TRECVID 2003 corpus and search topics, and the three methods performed comparably on our internal experiments. The ranked lists generated by the three approaches were therefore fused using a non-weighted query-independent Round Robin fusion—e.g., min rank aggregation of individual rank lists.

5.1.2 Visual retrieval

For our visual retrieval subsystem, we used the approach from [NNT05b]. For brevity, we give only a brief summary of the approach here along with main deviations from [NNT05b].

Visual retrieval using SVM. Discriminative modeling, such as Support Vector Machines (SVM), works well for the Concept Detection task. The use of the SVM for the Search Task faces two challenges:

- Very small number of distinct positive examples
- No negative examples

We overcome these challenges by creating pseudo-negative samples from unlabeled data and using a "bagging" approach to mitigate the problem of imbalanced learning. This allows us to apply SVM discriminative modeling to query topics with very limited training data (e.g., on the order of 10 positive examples only). Discriminative modeling is very complementary to other popular approaches for content-based retrieval, such as nearest-neighbor modeling. While nearest-neighbor approaches have good recall, they usually suffer from poor precision, which is where discriminative models, such as SVMs, can excel. We therefore explored the combination of SVM with the MECBR modified nearest-neighbor approach. In our experiments we confirmed that the two approaches are very complementary, each outperforming the other in some cases, with the combination outperforming both in almost all cases. The overall SVM topic modeling approach—and its combination with the MECBR approach—is illustrated in Figure 9. More details on the general approach can be found in [NNT05b] so we list only specific implementation decisions here.

Performance of SVM classifiers can vary significantly with variation in parameters of the models. To minimize sensitivity to these design choices, we use the variance and the trade-off parameters determined for each feature for the TRECVID 2005 Concept Detection task. Radial basis function (RBF) kernels usually perform better than other kernels, so our reported results use SVM with RBF kernel. The other two parameters, ratio between positive and negative examples and number of learned SVM hyperplanes (

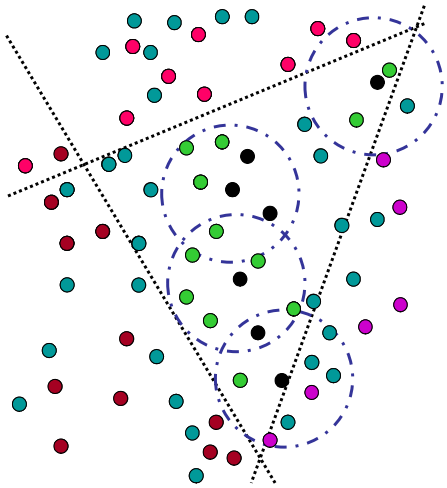

Figure 9: Combination Hypothesis Illustration. Each line represents a primitive SVM hyperplane between the same set of positive examples and a randomly sampled bag of pseudo-negative examples. Each circle represents a single CBR query executed by the MECBR method.

i.e. *bags*, as in [NNT05b]) are very specific to the nature of the search task, and therefore, optimized jointly. The SVM scores corresponding to each hyperplane are fused using AND logic, so that the final SVM model corresponds to the intersection of several positive hyper-spaces derived from each of the primitive SVM models. Thus, the objective of the ratio and bag number selection was to minimize the under-sampling rate of negative examples while avoiding the imbalance problem in the learning process [Jap00], and conserving the underlying distribution of the data points. These parameters were jointly determined based on the limiting factor of having a *very* small number of distinct (i.e. not near-duplicate) positive examples. Let N be the number of *bags*, K the imbalance ratio, and P number of positive examples for a specific query. We tested three different approaches for the parameter selection with initial K set to 10:

- Randomly sample the dataset for $N * K * P$ points that are not in the ϵ-neighborhood of positive examples. Pick $K * P$ points for each primitive SVM run.
- Randomly sample the dataset for $N * K * P$ points that are not in the ϵ-neighborhood of positive examples. Cluster the selected points using k-means algorithm into N clusters (with min number of items in each cluster set to $2 * P$), and run the primitive SVM run using each cluster as a pseudo-negative bag. K will vary for each cluster. Vary N.
- Cluster the dataset points using k-means algorithm into N clusters (with min number of items in each cluster set to P). Remove the clusters where there are more

than 10% of positive examples. Pick $K * P$ examples from one cluster at the time (initial N varies for the query topic), and use them as a pseudo-negatives in a primitive SVM run.

These approaches were tested for K=10, and N between 10 and 50 for the TRECVID 2005 Search Task testing on the development set. The first and the third method gave comparable results, while the second method underperformed. This is not surprising, since using pseudo-negatives from only one cluster can actually enable low selectivity in a high-dimensional feature space. We used the first approach for the final SVM-based visual runs due to its simplicity and consistent performance.

Visual retrieval using MECBR. We used the multi-example content-based retrieval (MECBR) approach from [NS03b, NNT05b] (see also Section 4.2.4) as a complementary approach to SVM for query topic modeling. While SVM works great when there is a sufficient number of training examples, including negative examples, it can learn only simple decision boundaries when given very few positive and some pseudo-negative examples. The decision hyperspace learned by SVM can therefore be refined through a combination with a nearest-neighbor type of classifier modeling the immediate neighborhood of the positive examples. The idea is that we would like to combine the recall of nearest-neighbor classifiers, such as MECBR, with the precision of discriminant classifiers, such as SVM. The approach is illustrated in Figure 9.

For more details of the two methods and the way the combination hypothesis works, please see [NNT05b]. The MECBR method used in the search task was identical to the one described in the above paper and also to some extent in Section 4.2.4. Specifically, unlike the concept detection task, in the search task we did not use any clustering or sampling of the positive examples since their number was limited to begin with. The complete set of positive visual examples for each topic included all image examples, if given, as well as up to 3 frames extracted from each of the given video segments. These frames were uniformly spaced within the segment after stripping the 5 boundary frames on both sides of the clip. Each example was then used independently as a CBR query and results were fused using OR logic (i.e., MAX aggregation of confidence scores). Other parameters (e.g., score normalization) were fixed globally on a feature-dependent but query-independent basis. The MECBR approach was used with four features—global color correlogram, color moments grid, global co-occurrence texture, and semantic model vectors.

Fusion. The final visual-only run was a combination of SVM-based and MECBR-based runs for 4 different features—color correlogram, color moments, co-occurrence texture, and semantic model vectors. These 8 runs were fused using simple non-weighted averaging of statistically normalized scores (e.g., see [NNT05b]). The only twist was that all low-level features were fused first, and the resulting ranked list based on visual features was then fused to the one based on semantic features. This was done so as to avoid bias towards the visual feature runs, which outnumbered the two semantic feature runs.

5.1.3 Model-based retrieval

Model-based retrieval applies the results from off-line concept detection and text analysis to on-line queries by triggering concept models with different weights. We have devised a supervised and an unsupervised method for off-line index creation, both producing text-to-model correlation indices that are used in the on-line query execution to produce model-based rankings of shots. Both methods are data-driven and use corpus co-occurrence statistics to compute a mapping from terms in the speech transcript to names of available models. The two approaches are illustrated in Figure 10.

The supervised index creation step leverages a standard text search engine to produce the text-to-models correlation index. We generate pseudo-documents for all ASR terms, after pre-processing the text with a stemmer and a phraser. Given a shot and its corresponding ASR text, we extend the pseudo-documents with each term's co-occurring text. We also add concept models from ground-truth concept annotations for the given shot to all of the pseudo-documents. A text search engine indexes the documents and computes TF-IDF statistics for each term. We refer to the resulting index as a text-to-model correlation index.

Our unsupervised approach involves the creation of a text-to-model correlation matrix. After pre-processing ASR text with a stemmer and a phraser, we weight ASR terms with models from automatic concept detection across all shots. The resulting correlations between ASR terms and corresponding concept models are aggregated in a matrix of terms to concepts. We refer to this resulting matrix also as a text-to-model correlation index.

Queries are evaluated in an on-line step, which leverages one of the off-line correlation indices to produce a model-based ranking of shots. Queries are analyzed in the same manner as ASR text in the previous steps; a stemmer and a phraser are used to resolve terms for matching query text to text-to-model correlation indices. In case of the supervised approach, the query text is evaluated using the search engine, resulting in a confidence list of best-matched models and corresponding weights proportional to the returned search engine scores. In the unsupervised approach, model confidences are selected from the text-to-models correlation index for each query term and are then fused across query terms to produce overall query-model correlation weights.

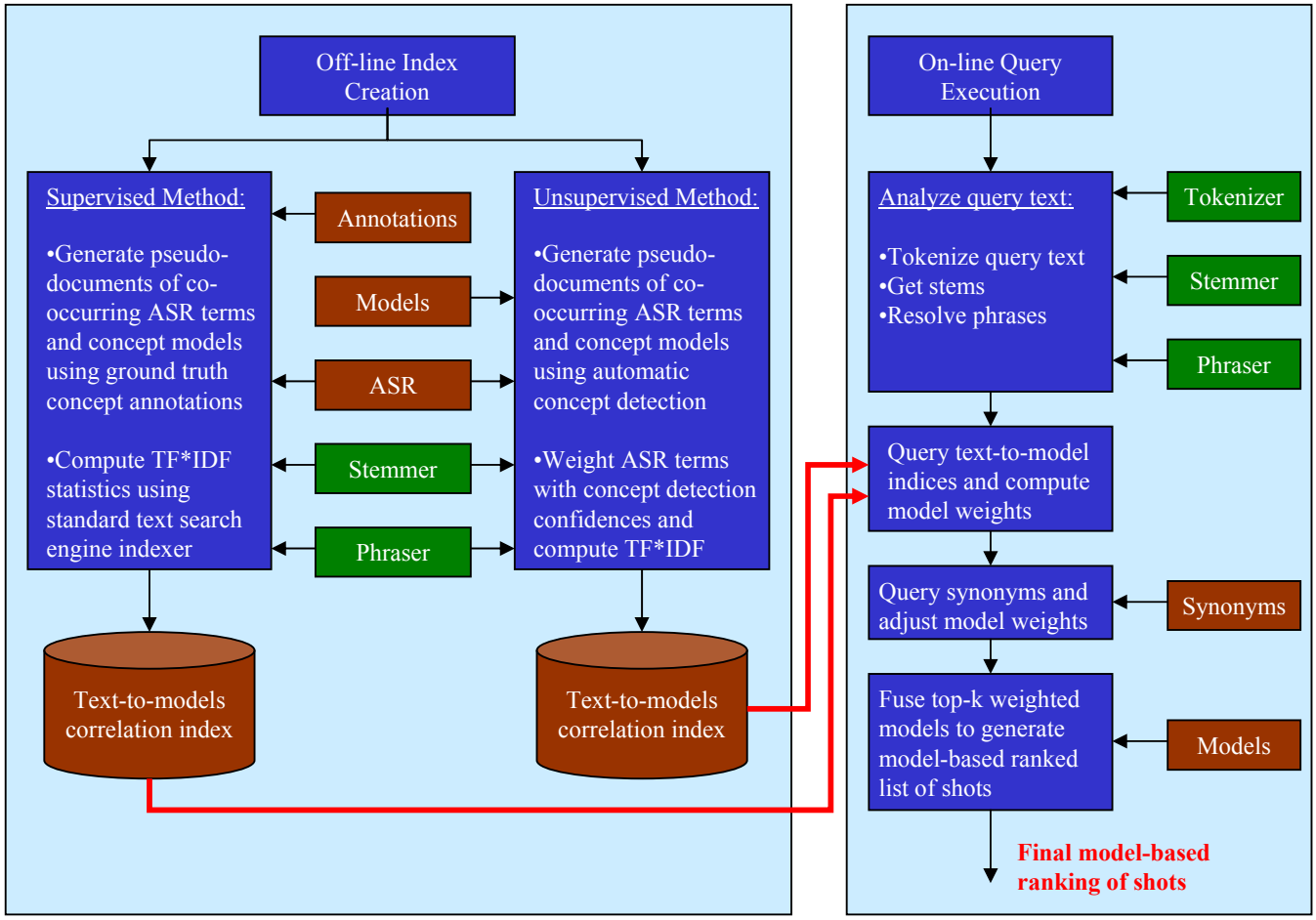

Figure 10: Overview of the model-based retrieval method, which applies the results from off-line concept detection and text analysis to on-line queries by triggering concept models with different weights. The approach consists of an off-line index creation phase and an on-line query expansion phase.

We then adjust and re-rank the resulting list of models and confidences for both approaches by evaluating the query against a list of synonyms for models and boosting the weights of models that were hit by the query. Given the final list of query-relevant models and weights, we select the most relevant k models with their corresponding weights, and fuse them using weighted average score aggregation to generate a final ranked list of shots. The resulting model-based ranked list is then used to re-rank lists from other retrieval methods through an appropriate fusion method.

5.1.4 Multimodal Fusion and Reranking

Our fusion approach for the search task was similar to that of the concept detection task (see Section 4.3). However, due to the lack of a training set for fusion parameter tuning, we had to resort to globally tuned query-independent fusion, as opposed query-class-dependent fusion methods as in [KNC05]. In particular, we used the following rules when fusing multiple runs from the same modality or from different modalities:

- Fusion of visual runs only—we used non-weighted score averaging of statistically normalized scores.
- Fusion of text runs only—we used non-weighted score averaging of rank normalized scores.
- Fusion of text and visual runs—we used round robin fusion by doing rank normalization followed by MAX score aggregation.
- Model-based re-ranking of text runs—we used non-weighted score averaging of rank normalized scores.
- Model-based re-ranking of visual runs—we used non-weighted averaging of statistically normalized scores.
- Fusion of text + visual + models—same as text+visual fusion but using the model-based re-ranked text and visual runs as inputs.

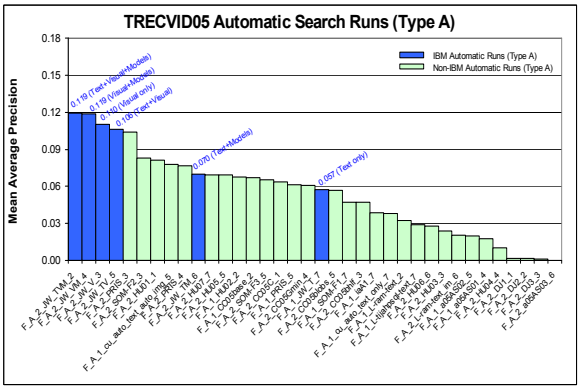

Figure 11: Mean Average Precision performance of IBM automatic search type A runs relative to other automatic type A submissions.

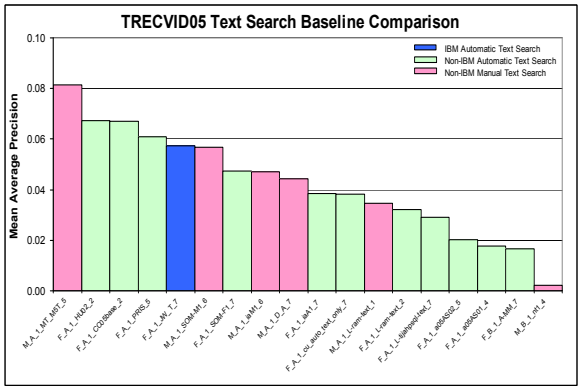

Figure 12: Mean Average Precision performance of automatic and manual text-only required baselines.

5.1.5 Experiments and Results

We submitted 6 automatic type A runs for this year's Search Task, which are listed with their corresponding MAP scores in Table 3 and in Figure 11.

Run ID	Run Description	MAP
F_A_1_JW_T_7	Text	0.057
F_A_2_JW_TM_6	Text + Models	0.070
F_A_2_JW_TV_5	Text + Visual	0.106
F_A_2_JW_V_3	Visual	0.110
F_A_2_JW_VM_4	Visual + Models	0.119
F_A_2_JW_TVM_2	Text + Visual + Models	0.119

Table 3: Mean Average Precision scores for all IBM automatic search submissions.

For speech-based retrieval, we used only the required ASR/MT transcripts and experimented with several different approaches for automatic query refinement. These included query expansion based on pseudo-relevance feedback, lexical affinities, as well as automatic detection and expansion to over 100 semantic categories developed originally for question answering purposes. While our text-based run performed competitively relative to other text-only submissions (it was 4th out of 11 comparable submissions and well above the mean and the median)—we generally found the text modality to be much less reliable this year due to the effects of machine translation for foreign sources. This run had our lowest Mean Average Precision of 0.057 and its performance relative to all automatic and manual text-only baselines is shown in Figure 12.

The highlight of our system this year was our visual-only run, which performed nearly twice as well as our text-only baseline (MAP of 0.110), and outperformed all automatic type A submissions by other participants as well as 26 of the 28 manual submissions. For this run, we explored a novel approach leveraging two complementary light-weight learners—modified nearest-neighbor and SVMs—to solve the problem of visual query modeling with few examples. The details of this approach can be found in [NNT05b].

Another approach that worked very well for us was model-based re-ranking and expansion of result lists generated by speech-based and/or visual-based retrieval. We build SVM models for all 39 concepts that were annotated, and used these models to generate a rank list for each query based on how relevant each model was to the query text. We measured query-to-model relevance using two approaches, combing corpus-based co-occurrence of ASR terms and models, along with model synonym-based query expansion. The final model-based ranked result list was then fused with ranked lists generated by other retrieval methods so that the other lists are effectively re-ranked based on the weights of the most relevant models for each query. Model-based re-ranking improved upon the text-only baseline by over 20% to a MAP of 0.070. It also lead to our top 2 runs (both with equal MAP of 0.119) by re-ranking our visual-only and text+visual runs, improving them by 8% and 12%, respectively.

5.2 Interactive Search

The IBM Marvel Multimedia Analysis and Retrieval System was used for our interactive search run. Marvel provides search facilities for content-based (features), model-based (semantic concepts) and text-based (speech terms) querying.

Marvel allows users to fuse together multiple searches within each query, which was typically done for answering the TRECVID query topics. For example, given the statement of information need and query content, the user would typically issue multiple searches based on the example content, models and speech terms. In many cases, the results from an automatic run were used to kickoff the interactive

Figure 13: Marvel multimedia analysis and retrieval system used for interactive search (results for qry158).

search. Figure 13 illustrates the Marvel multimedia analysis and retrieval system. An on-line demo of the system can be accessed from http://www.research.ibm.com/marvel/.

6 Observations and Future Directions

We experimented with a number of new techniques in the context of TRECVID 2005. For the detection task, we observed robust performance with the baseline support vector learner and were able to improve upon its performance using an ensemble of other generic learning techniques resulting in top performance. For the search task, we were able to automatically leverage for the first time, models as well as model vectors along with our existing bank of features. We were also able to leverage for the first time through the IBM UIMA Text Analytics components, various text analytics and processing functionalities that when combined with our low-level and high-level visual features and models resulted in the top performing automatic Type A search runs. We are working to improve upon the new directions including better leveraging of models as well as learning models across granularities. We also plan to work on improving our interactive search capabilities so that it can fully and efficiently expose our analytical capabilities at search run time with minimal onus on the human in the loop.

References

[ACF+04] A. Amir, S.-F. Chang, M. Franz, G. Iyengar, J. R. Kender, C.-Y. Lin, M. R. Naphade, A. Natsev, J. R. Smith, and J. Tešić. IBM Research TRECVID-2004 video retrieval system. In *NIST Text Retrieval Conference (TREC)*, Gaithersburg, Maryland, November 2004.

[AHI+03] A. Amir, W. Hsu, G. Iyengar, C.-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. J. Nock, J. R. Smith, B. Tseng, Y. Wu, and D. Zhang. IBM Research TRECVID-2003 video retrieval system. In *NIST Text Retrieval Conference (TREC)*, Gaithersburg, Maryland, November 2003.

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

[FL04] D. Ferrucci and A. Lally. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Journal of Natural Language Engineering*, 10(3-4):327–348, 2004.

[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):245–268, Dec 1999.

[J. 05] J. Argillander, G. Iyengar and Harriet Nock. Semantic Annotation of Multimedia Using Maximum Entropy Models. *ICASSP*, 2005.

[Jap00] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *International Conference on Artificial Intelligence: Special Track on Inductive Learning*, Las Vegas, Nevada, June 2000.

[Jay57] E. T. Jaynes. Information theory and statistical mechanics. *The Physical Review*, 106:620–630, 1957.

[JM04] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. In *Image and Video Retrieval: Third International Conference, (CIVR)*, Lecture Notes in Computer Science, Dublin, Ireland, July 2004. Springer-Verlag.

[KNC05] L. Kennedy, A. Natsev, and S.-F. Chang. Automatic discovery of query-class-dependent models for multimodal search. In *ACM Multimedia 2005*, Singapore, Nov. 2005.

[MLP98] O. Maron and T. Lozano-Perez. A framework for multiple instance learning. In *Neural Information Processing Systems*. MIT Press, 1998.

[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.

[MSS02] B.S. Manjunath, Philippe Salembier, and Thomas Sikora, editors. *Introduction to MPEG–7: Multimedia Content Description Interface*. John Wiley & Sons Ltd., June 2002.

[mst]

[NBS+02] M. Naphade, S. Basu, J. Smith, C. Lin, and B. Tseng. Modeling semnatic concepts to support query by keywords in video. In *IEEE International Conference on Image Processing*, Rochester, NY, Sep 2002.

[NJ03] M. Naphade and J.Smith. The role of classifiers in multimedia content management. In *SPIE Storage and Retrieval for Media Databases*, volume 5021, San Jose, CA, Jan 2003.

[NKK+] Milind Naphade, Lyndon Kennedy, John Kender, Shih-Fu Chang, John R. Smith, Paul Over, and Alexander Hauptmann. A light scale concept ontology for multimedia understanding for TRECVID 2005, ibm technical report RC23612.

[NLN+03] M. Naphade, C. Lin, A. Natsev, B. Tseng, and J. Smith. A framework for moderate vocabulary visual semantic concept detection. In *Proc. IEEE International Conference on Multimedia and Expo*, July 2003.

[NNT05a] A. Natsev, M. Naphade, and J. Tesic. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM Multimedia 2005*, Singapore, Nov 2005.

[NNT05b] Apostol Natsev, Milind R. Naphade, and Jelena Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *ACM Multimedia*, Singapore, November 2005.

[NS03a] Milind R. Naphade and John R. Smith. A hybrid framework for detecting the semantics of concepts and context. In M. Lew, N. Sebe, and J. Eakins, editors, *Lecture Notes in Computer Science: Image and Video Retrieval*. Springer, 2003.

[NS03b] A. Natsev and J. R. Smith. Active selection for multi-example querying by content. In *IEEE Intl. Conf. on Multimedia and Expo (ICME '03)*, Baltimore, MD, July 2003.

[NS03c] A. Natsev and J. R. Smith. New anchor selection methods for image retrieval. In *Proc. SPIE Electronic Imaging: Storage and Retrieval for Media Databases*, San Jose, CA, Jan. 2003.

[NS05] M. Naphade and J. Smith. A generalized multiple instance learning algorithm for large scale modeling of multimedia semantics. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Philadelphia, PA, May 2005.

[NSS04] Milind Naphade, John Smith, and Fabrice Souvannavong. On the detection of semantic concepts at TRECVID. In *ACM Multimedia*, New York, NY, Nov 2004.

[PBCR00] J. Prager, E. Brown, A. Coden, and D. Radev. Question-answering by predictive annotation. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 184–191, Athens, Greece, 2000.

[Pet] C. Petersohn. Fraunhofer HHI at TRECVID 2004: Shot boundary detection system. TREC Video Retrieval Evaluation Online Proceedings. Available at http://www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc., Englewood Cliffs, NJ, 1971.

[SNN03] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *IEEE ICME 2003*, Baltimore, MD, July 2003.

[uim] UIMA SDK. Available at IBM Alphaworks, http://www.alphaworks.ibm.com/tech/uima.

[Vap95] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[vtr02] TREC Video Retrieval, 2002. National Institute of Standards and Technology, http://www-nlpir.nist.gov/projects/trecvid/.

[XC96] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, New York, NY, 1996.