# IBM Research TRECVID-2006 Video Retrieval System

Murray Campbell[*], Alexander Haubold[†], Shahram Ebadollahi[*], Dhiraj Joshi[*], Milind R. Naphade[*], Apostol (Paul) Natsev[*], Joachim Seidl[*], John R. Smith[*], Katya Scheinberg[*], Jelena Tešić[*], Lexing Xie[*]

## Abstract

*In this paper, we describe the IBM Research system for indexing, analysis, and retrieval of video as applied to the TREC-2006 video retrieval benchmark. This year, focus of the system improvement was on ensemble learning and fusion for both high-level feature detection task and the search task.*

*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 High-level Feature Detection, Search tasks, and Rushes experimental task. In this paper, will describe the IBM Research system and examine the approaches and results for the all three 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.

Our high-level feature detection system benefited from multiple learning approaches and learned fusion. This year we used consider different random partitions of training and internal validation sets to build several SVM models for all concepts over all features. We also considered multiple views of the ground truth itself where more than one annotator input exists for the development corpus. Multi-kernel linear machines provided an interesting context for fusion across features at the kernel level for rare concepts in 39 LSCOM-lite set. Fusion over such different views, models and methods resulted in 22 % average improvement over visual baseline.

We developed a fully automatic retrieval systems for speech, visual and semantic modality, and produced the top runs among automatic type A search systems. We used a new text search engine for our speech-based retrieval system and explored multiple automatic query refinement methods for it. For our visual and semantic retrieval systems, we applied a light weight learning approach. This year, our main focus was on the multimodal fusion component of the system for combining our speech, visual, model-based and semantic runs. We have explored query-dependent search fusion among the text, model, and visual retrieval scores. Our two query-class-dependent fusion approaches resulted in top two performance runs with 0.0855 and 0.086708 MAP respectively. Query dependant fusion gain was around 13% compared to simple query-independent non-weighted fusion method run. Overall, our improved speech, semantic and visual approaches and query dependant fusion approaches were the key performance contributors for our system.

For the rushes task, we have improved our existing search system and extended the list of functionalities to easily browse through data collection using different modalities: metadata, visual, concept, and tags.

## 2 Video Descriptors

### 2.1 Visual Features

The system extracts eight different visual descriptors at various granularities for each representative keyframe of

[*]IBM T. J. Watson Research Center, Hawthorne, NY, USA
[†]Dept. of Computer Science, Columbia University

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.

Last year, we performed extensive experiments using the TRECVID 2005 development set 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 consistent top performance for both search and concept modeling experiments:

- 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.

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. We use the term *visual-based approach* to denote search methods in low-level visual descriptor space.

## 2.2 Semantic Feature

The Large-Scale Concept Ontology for Multimedia (LSCOM) is a first of its kind effort, designed to simultaneously optimize utility, to 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]. LSCOM-lite is a subset of 39 concepts from the full LSCOM taxonomy and was jointly annotated by the TRECVID community in 2005, see Figure 1. The semantic-based retrieval approach presented in this work relies on a previously modeled high-level descriptor space, which for the purposes of Hig level feature detection task consists of the 39 LSCOM-lite concepts. We apply concept detection to the query examples and generate model vector features consisting of the confidences of detection for each of the concept models in our lexicon (e.g., a 39-dimensional feature vector based on the LSCOM-lite lexicon) [NNS04]. These features are then used just like any other content-based features and retrieval is performed by the same light-weight learning methods used for visual retrieval. We use the term *Semantic space* to denote a vector space comprised of model scores as a feature descriptor space for search, and the term *semantic-based approach* is used to denote search methods in semantic spaces.

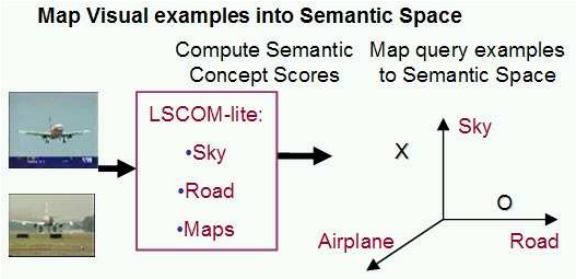



Figure 1: Semantic feature extraction from LSCOM-lite model classification scores

## 2.3 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.

## 2.4 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:

1. Term Frequency (TF) in given shot document
2. Inverse Document Frequency (IDF) across all shot documents
3. TF$\times$IDF
4. 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 $i^{th}$ dimension reflected one of the above measures for the $i^{th}$ term in the speech vocabulary. These features were used for SVM-based modeling in the High level feature detection task.

# 3 High-level Feature Detection

## 3.1 Support Vector Machine Ensembles for Improving Performance

Figure 2(a) illustrates the IBM high level feature detection system. 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 [NNT05, NSS04, NBS$^+$02]. Our low-level visual features are 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.

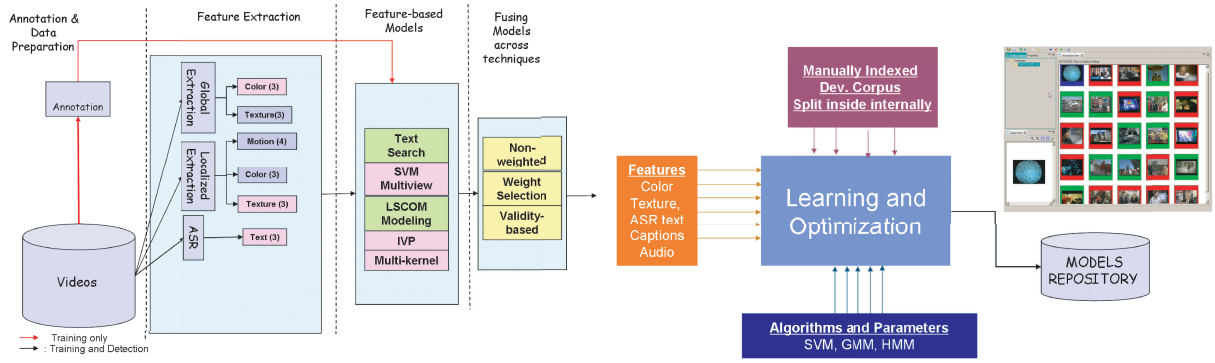
### 3.1.1 Data Partitioning

We partitioned the development data set provided by NIST into the following 3 internal partitions for facilitating hierarchical processing experiments and selection by randomly assigning videos from the development set to each partition. The table 1 below gives the number of keyframes in each partition for models in 2005 and 2006. We used different partitioning for TRECVID 2005 and TRECVID 2006 training, and we leveraged both to build final models in 2006.

| **Models (year)** | Training | Validation | Fusion |
|---|---|---|---|
| 2005 | 41K | 7K | 7K |
| 2006 | 45798 | 10865 | 5238 |

Table 1: Data partitioning of the development set used to build TRECIVD 2005 and TRECVID 2006 models. TREVCVID 2005 partition has a selection set 7K for fusion optimization.

Figure 2(b) illustrates the modeling and optimization approach. This year we tried to go two steps further. One was to also consider different random partitions of training and internal validation sets to build several additional models for all concepts over all features. These models then get combined using naive fusion strategies during detection and fusion. In addition to considering various




(a) System Overview (b) Modeling Tool

Figure 2: The IBM 2006 TRECVID High-level Feature Detection System (a) overview, and (b) modeler component for annotation and model building. Model building component handles data partitioning, parameter optimization, and cross validation with multiple optimality criteria.

views of the development data set through multiple partitions and models derived from those, is to take multiple views of the ground truth itself where more than one annotator input exists for the development corpus. This second additional dimension leads to further model building based on various automatic interpretations of the ground truth. The various interpretations are derived by automatically fusing the multiple annotations for the development corpus wherever they exist using fusion operators such as max, min, average, etc. The actual model building is performed using the IBM Marvel Modeler tool (a screenshot of its annotation interface can be seen in Figure 2(b)) which automates everything including the partitioning, and feature and parameter optimization under the hood thus creating a simple interface for non-experts who want to build good quality models based on several best practises that we have developed over the past five years of the benchmark.

**Additional LSCOM models built for Type B System** Due to time limitations we were unable to build models for all the LSCOM concepts [NST+06] but we confined ourselves to a small set of models that we thought could be relevant to the 39 LSCOM-lite concepts being detected [OIKS]. The mapping and relevance weights of the LSCOM concepts for the LSCOM-lite concepts, was done manually, and included in our one Type B submission.

## 3.2 Multiple Kernel Learning

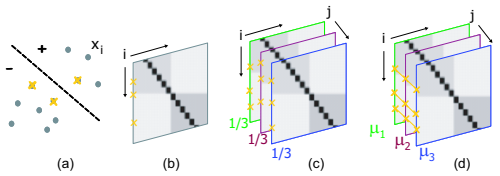



Figure 3: Learning class discrimination with multiple kernels. Yellow crosses ($\times$) denotes support vectors; red, green and blue denotes different kernels and their weights. (a) Linear classifier in the feature space. (b) A single SVM, or averaging kernels. (c) Averaging multiple SVMs. (d) Multiple Kernel Learning with shared support vectors and learned kernel weights.

In visual recognition applications we often have more than one type of cues from the data. They can come in the form of different types of descriptors, such as color-correlogram or semantic concepts, or in the form of different types of feature design from common features, such as the choices for modeling time and computing similarity in Sections 2 and prior work [EXCS06]. Two questions naturally arise: (1) Can we collectively use these multiple cues to make better prediction of the concept? (2) Can we simultaneously learn the importance of each of the input cues?

We consider multiple cue fusion in the context of SVM-like kernel classifiers, i.e., linear fusion for learning a lin-

ear discriminant in a high-dimensional feature space as shown in Fig. 3(a). Denote the pool of training shots as $v_i$, $i = 1, ..$, the collection of $k$ different kernels as $K_j(\cdot, \cdot)$, $j = 1, ..k$. There are several popular practices for this task [TLN+03]. Fig. 3(b) depicts "early-fusion", i.e., concatenating input vectors or averaging the different kernel values to arrive at a single kernel $\bar{K}(v_i, \cdot)$, and then learn a single SVM for class separation. Denote the support vector weights as $\alpha_i$, the decision function for a test example $\hat{v}$ is then written as

$$\hat{y} = \sum_i \alpha_i \bar{K}(v_i, \hat{v}). \quad (1)$$

Fig. 3(c), nick-named "late-fusion", corresponds to learning $k$ SVMs independently and average the decision values, with $\alpha_{i,j}$ the kernel-specific support vector weights, in this case the decision value is computed as in Equation (2).

$$\hat{y} = 1/k \sum_j \sum_i \alpha_{ij} K_j(\hat{x}, x_i). \quad (2)$$

These fusion schemes has two notable drawbacks: (1) neither take into account the relative importance among different kernels, (2) the "late fusion" requires $k$ rounds of training for different SVMs, leading not only to increased computational requirements in training time, but also a larger trace of the model that increases the classification time and memory requirements. It is also possible to learn another layer of SVM for kernel weights on the decision values from the individual SVMs, however this not only increases the computational complexity, but also needs to stratify the training data and is more prone to over-fitting.

To complement the existing fusion schemes in these two aspects, we explore the Multiple Kernel Learning (MKL) decision function in the form of Equation (3) and Fig. 3(d) for multi-cue fusion in visual recognition, i.e., learning linear weights $\mu_j$ among the kernels $j = 1, ..k$ with shared support vector weights $\alpha_i$.

$$\hat{y} = \sum_j \sum_i \mu_j \alpha_i K_j(\hat{x}, x_i) \quad (3)$$

Proposed recently by Bach and Jordan [BLJ], this decision function can also be viewed as one SVM with support vector weights $\alpha_i$ over a "hyper-kernel" $\sum_j \mu_j K_j(\cdot, v_i)$. Compared to the early and late-fusion schemes, the number of parameters of MKL is close to those of the early fusion, and the set of kernel weights naturally lends to interpretations of the result.

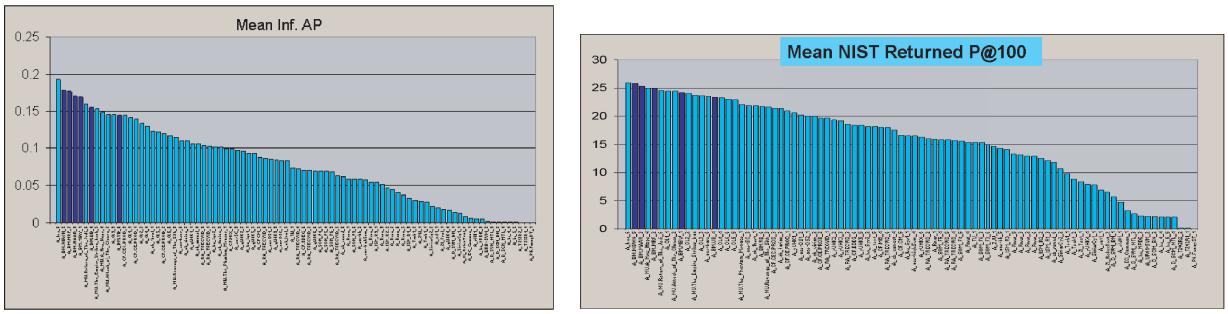
It is shown [BLJ] that this problem can be formulated in its dual form as Problem (4), i.e., solving for optimal nonnegative linear coefficients $\mu_j \geq 0$ so that the trace of $\sum_{j=1}^k \mu_j K_j$ remains constant (chosen to be equal to $d = \mathrm{tr}(\sum_{j=1}^k K_j)$) and so that the soft margin SVM is optimized with respect to this linear combination of the kernel matrices.

$$\begin{aligned} \min \quad & \frac{\gamma^2}{2} - e^\top \lambda \quad (4) \\ \text{s. t.} \quad & \lambda^\top D_y K_j D_y \lambda \leq \frac{\mathrm{tr}(K_j)}{d} \gamma^2 \quad j = 1, \ldots, k \end{aligned}$$

where $D_y$ is the diagonal matrix with the labels $y$ on the diagonal and $C$ is the soft margin penalty parameter determined with cross-validation. This problem can in turn be converted into a standard form of second-order-cone programming, and we obtain its solutions with the convex solver Sedumi [Stu99].

## 3.3 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 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 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.




(a) mean AP

(b) precision@100

Figure 4: Retrieval performance of IBM high level feature runs in context of (a) all the Type A submissions using the new mean inferred average precision measure (b) all the Type A submissions using the mean performance of precision achieved at a depth of 100

**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.

To generate the runs, we performed detection over the concepts first using the following individual approaches and then proceeded to fuse the resultant retrieval lists with described normalization and fusion techniques.

1. SVM-2005: SVM Models built during TRECVID 2005 for all 39 concepts using the 2005 data partitions and single interpretation of the ground truth
2. SVM-2006: SVM Models build for TRECVID 2006 using IBM Marvel Modeler for all 39 concepts using a new partitioning of the development corpus with varying interpretations of the groundtruth
3. Text: Text retrieval for all 39 concepts
4. LSCOM: To enforce context selectively we built additional concept models beyond the required 39 using LSCOM annotations. These models were used to leverage context for the following 4 concepts: *Boat/Ship, Car, Government Leader and Waterscape/Waterfront*
5. MKM: Multi-kernel linear models for 4 concepts: *Bus, Court, Natural Disasters and Snow*
6. IVP: Image Upsampling based SVM model for 1 concept: *Animal*

## 3.4 Submitted Systems and Results

Based on all the experiments we submitted the following 6 runs:

If the mean inferred average precision is to be considered as a measure of the overall performance of the systems submitted, it can be seen that most of the runs appear to have similar performance except for the visual only baseline, see Figure 4. A selection strategy between the visual versus text based retrieval based on performance on the held out set improves performance over visual only

| Run | Priority | Type | Description | MAP |
|---|---|---|---|---|
| VB | 6 | A | Naiive fusion of SVM-2005 and SVM-2006 | 0.145 |
| UB | 4 | A | Best of Naiive Fusion of SVM-2005 and SVM-2006 or Text | 0.156 |
| MBW | 1 | A | Gaussian normalization and Weighted Fusion of SVM-2005, SVM-2006 and Text | 0.169 |
| MBWN | 5 | A | Sigmoid Normalization and Naiive Fusion of SVM-2005, SVM-2006 and Text | 0.177 |
| MRF | 2 | A | Weighted fusion of SVM-2005, SVM-2006,Text, MKM and IVP | 0.176 |
| MAAR | 3 | B | Weighted fusion of SVM-2005, SVM-2006,Text, MKM, IVP and LSCOM Context Models | 0.17 |

Table 2: IBM TRECVID 2006 High level Feature Detection Task – Submitted Runs

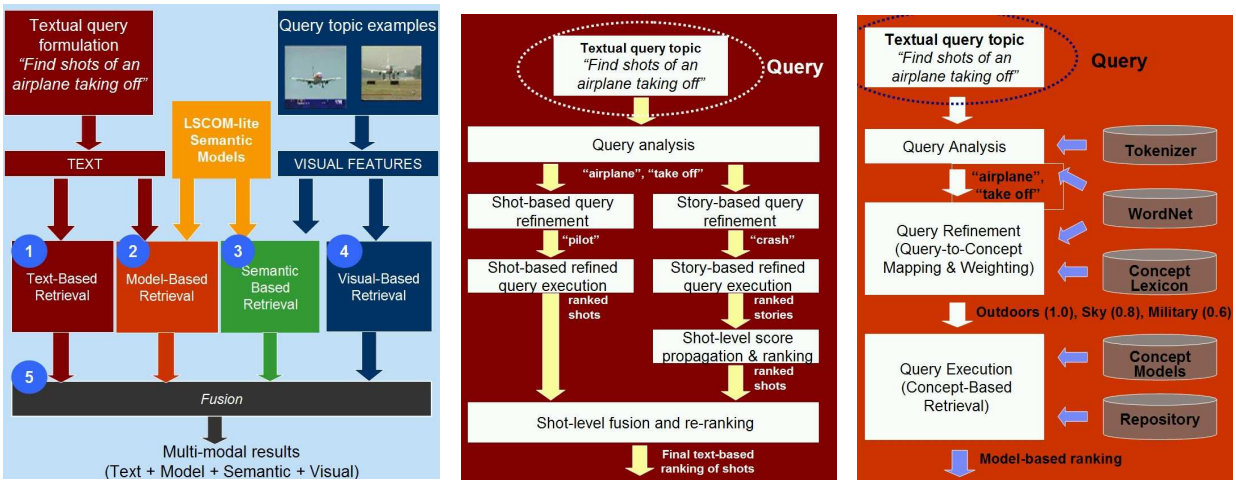
detection by 7 %. Fusing across the two modalities using our SVM-2005 and SVM-2006 visual models and text baseline results in an improvement of 17 % with weights derived from a held out set and gaussian normalization prior to fusion. When a sigmoidal normalization scheme is employed with naive fusion the performance over the visual only baseline improves by 22 %. Note: It was noticed based on internal experiments that the actual precision at 100 for each of the six IBM runs was double that of the number reported by NIST. This discrepancy is assumed to be on account of the sampling that was performed prior to evaluation.

Some concepts benefit significantly by the multi-modal fusion. For example *Airplane* performance jumps up from a mere 3.6 % for the visual only baseline to 16.6 % for the multimodal fusion across visual svm results and text although the text alone is not any better than the visual alone. This indicates reranking and improvement in precision when the two modalities are fused. A further improvement in performance also can be seen for some concepts with context fusion. For example *Airplane* improves from 16.6 % AP to 21 % AP when fused with the LSCOM context models of concepts related to airplanes such as airplane taking off, airplane landing, airport etc. Similar improvement is also seen in the case of the concept *Car* whose performance improves from 16.5% with visual SVM detection to 19.6 % with multimodal to 21% with context fusion using concepts such as vehicle, road, etc. Improvement however was not observed for *Waterscape*, the third concept for which we used context. The other concepts for which we used context were not evaluated. Newer techniques that we are also investigating including the image upsampling prior to modeling also help improve performance for 1 concept, *Animal*. The multi-kernel linear machines which provide an interesting context for fusion across features at the kernel level but the four concepts for which we used this idea were not among the twenty concepts evaluated this year.

# 4 Automatic Search

The IBM team continued its focus on automatic search for this year's TRECVID, submitting 5 automatic runs (type A). Two of our automatic runs outperformed all other automatic and manual runs in Mean Average Precision scores. The overall architecture of our automatic search system was again a combination of speech-based retrieval with automatic query refinement, visual retrieval based on light-weight learning, and model-based retrieval and re-ranking using automatic concept detectors for the 39 LSCOM-light concepts [OIKS] (see also system overview in Figure 5). Most processing was done at the sub-shot level based on the master shot boundary reference [OIKS], 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.

Changes in our speech-based retrieval (component 1 in Figure 5(a)) system this year included retrieval at the story level (for improved recall) with re-ranking at the shot-level (for improved precision), as well as improved




(a) System Overview (b) Speech component (c) Model-text component

Figure 5: Overview of IBM automatic search system and its components (a) overview off all system components, (b) Speech-based retrieval component, and (c) Model-based retrieval component using query text.

parameter tuning for automatic query expansion and re-ranking with the IBM Semantic Search engine (aka JuruXML) [MMA+02]. For our required baseline, we used only the common ASR/MT transcripts and our shot-level retrieval system had a MAP score of 0.041. Our improved speech-based retrieval system used the story boundaries donated by Columbia University [HC05], as well as speaker segmentation boundaries provided to us by the NUS team[OIKS], performed significantly better, generating a MAP score of 0.052, or nearly a 30% improvement over the baseline.

This year we significantly expanded our emphasis on model-based retrieval and re-ranking using automatic concept detectors for the 39 LSCOM-lite concepts. We experimented with several approaches for automatic query-to-model mapping (component 2 in Figure 5(a)) and weighting from query text, including the lexical and statistical approaches we tried last year, as well as a new rule-based ontology mapping approach, resulting in the best MAP of 0.029.

Our semantic-based run (component 3 in Figure 5(a)) is interpreting semantic space from Section 2.2 as a descriptor space. Our visual (component 4 in Figure 5(a)) and semantic retrieval system were an improved combination of two light-weight learning algorithms — modified k-Nearest Neighbor classifier and SVM with pseudo-negative sampling and bagging. This year improvement can be contributed greatly to smarter and more robust data modeling techniques.

The final component of the IBM automatic search system was the emphasis on multimodal fusion (component 5 in Figure 5(a)). We tried out three different multimodal fusion approaches—a query-independent non-weighted fusion approach, and two query-class-dependent fusion approaches using strict and fuzzy query class assignments of the four components. These approaches generated our best runs with MAP scores of 0.076, 0.086, and 0.087.

## 4.1 Speech-based retrieval

Our speech-based retrieval system is shown in Figure5(b). It is based on the JuruXML semantic search engine [MMA+02], which is available in the Unstructured Information Management Architecture (UIMA) SDK [uima]. For our speech-retrieval baseline, 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]. Each sub-shot was first expanded on the left to include the 5 preceding sub-shots, and was aligned at the speaker or phrase boundaries for the purposes of speech transcript indexing.

In addition to the base UIMA SDK, we used several

UIMA components developed by IBM Research for advanced text analytic. 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 2005 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.

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 TRECVID05 test set, query topics, and NIST-pooled topic ground truth. This speech-only baseline run had a MAP score of 0.041.

In order to improve recall without sacrificing precision, we also considered indexing and retrieval at the news story level, with story boundaries automatically extracted and provided by Columbia University [HC05]. In that case, we aligned the raw story boundaries with the speaker/phrase boundaries, and for each story we generated a text document consisting of the corresponding ASR/MT transcript. At query time, we first retrieved relevant stories, as ranked by the JuruXML search engine, propagated the score for each relevant story to all sub-shots in the story, and then fused the results (using simple score averaging) with the shot-level baseline retrieval results in order to break ties within the same story and re-rank shots for improved precision. This run generated a MAP score of 0.052, which is a significant improvement of nearly 30% over the baseline.

## 4.2 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. Given an arbitrary text- or example-based 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 model-based query result list is then used to re-rank results generated from other retrieval methods through an appropriate fusion method. For all model-based retrieval purposes we used our detectors for the 39 LSCOM-lite concepts [NST+06]. When the query-to-concept relevancy is determined based on query text alone, we considered a lexical approach to text to model mapping. This is the same approach that we used last year at TRECVID [AAC+05] and it uses the WordNet-based Lesk similarity relatedness measure [BP03, PBP03] to compute the lexical similarity between the query text and the textual description for each concept model [HN06]. This approach results in the best overall MAP of 0.029, and it is illustrated in Figure 5(c).

## 4.3 Content-based Modeling

IBM TRECVID search visual and semantic based components are relying solely on query topic visual examples. Thus, the underlying retrieval approach is essentially the same for both components. We term it *content-based approach*. Content-based approach uses the unique approach of formulating the topic answering problem as a discriminant modeling one. The major improvement this year is in the area of data modeling.

Our baseline method, used in [AAC+05] combination hypothesis, fuses the selective MECBR (multi-example content based retrieval) approach with the discriminant SVM (support vector machines) one. Detailed baseline implementation is presented in [NNT05]. Figure 7 illustrates the basic idea. Circles show a a single CBR, and MECBR baseline is achieved using OR logic. SVM approach with nonlinear kernels allow us to learn nonlinear decision boundaries even when the descriptors is 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 is no negative examples provided, we generate pseudo-negative examples by randomly sampling data points. We build a set of primitive SVM classifier whereby the positive examples are used commonly across all classifiers but the pseudo-negative data points one from different sample set. The SVM scores corresponding to each primitive SVM model trained are then fused using AND logic to obtain a final discriminative model, as illustrated by the dividing lines in Figure 7. SVM-based search method proved to significantly improve the retrieval results over MECBR-based baseline approach, resulting in over 50% MAP improvement for the color modality [NNT05] over TRECVID 2003 search topics.

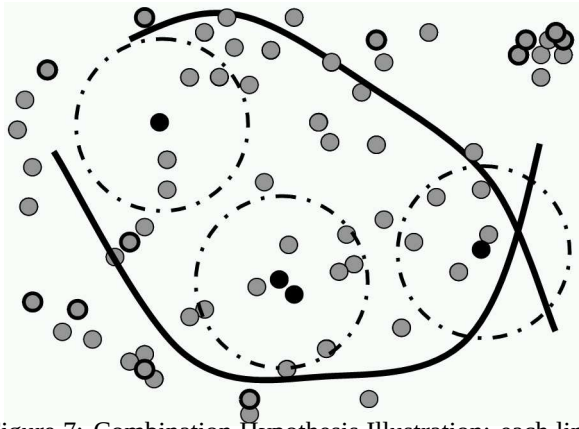
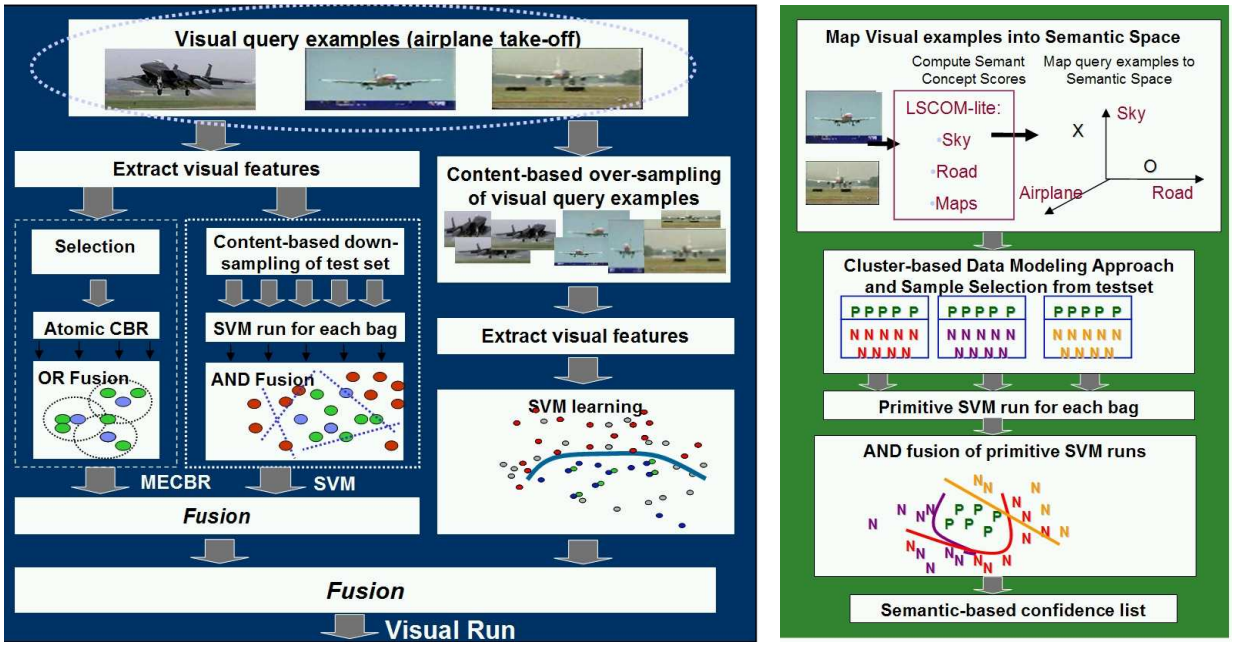


Figure 7: Combination Hypothesis Illustration: each line represents a primitive SVM hyperplane between the same set of positive examples (black fill) and a randomly sampled bag of pseudo-negative examples (black edge). Each dash-dot circle represents a single CBR topic.

### 4.3.1 Descriptor Space Modeling

For the video search experiments, we are faced with the limiting factor of having a *very* small number of distinct positive examples, and no negative examples. We overcome these challenges by (a) fusing a number of primitive SVM predictions trained on the same set of positives and different views of pseudo-negative selection data points so that the final SVM model corresponds to the intersection of several hyper-spaces, and (b) sampling pseudo-




(a) Visual-based component

(b) Semantic-based component

Figure 6: Overview of the content-based components of IBM automatic search system (a) Visual-based retrieval component, and (c) Semantic-based retrieval component using query topic examples.

negative data points so that they model the test space well. The objective here is to carefully select the pseudo-negatives to model the input space well, and to balance the number of pseudo-negative data points for training with number of positive examples to avoid the imbalance problem in the learning process [AKJ04]. The inherited objective is to maximize the number of selected pseudo-negative data points in the descriptor space. We propose to:

*maximize* the number of pseudo-negative data points under constraints of imbalanced learning and complexity, and

*carefully* select data points so that the descriptor space is well represented.

**Imbalanced ratio** In the SVM fusion framework of primitive models, we select $N$ pseudo-negative points for training from the targeted set, given $P$ positive external examples for sampling for each primitive SVM model, and $K$ primitive SVM models to be fused for final modeling. In selecting the number of pseudo-negative points $N$ for each primitive SVM model, the objective is to minimize the under-sampling rate of negative examples while avoiding the imbalance problem in the learning process, and therefore we need to maximize the ratio of negatives and positives rather than number of negatives alone. We adopted $N = 50$ as a fixed pseudo-negatives bag size in [NNT05]. To maximize the number of $N$ per model, we revisit this assumption here, and make $N$ a function of $P$ for every topic, where $P$ is the number of visual examples per topic. As reported in [AKJ04], maximum ratios should be less than 10 ($max\{N/P\} < 10$) so that SVM classifiers perform correctly. Descriptor space modeling using pseudo-negative data selection involves two stages: (a) sampling of the data points and (b) selection of the data points for each primitive SVM. We investigate two approaches to pseudo-negative sampling of $N \times K$ points from the dataset, ($max\{N/P\} < 10$):

**random** Select $N$ random points from the whole datastet for each of the $K$ primitive SVM models

**cluster** The core idea is to utilize supervised and unsupervised classification in concert in a light-weight learn-

ing process that generates smaller more effective models. To model the high-dimensional target space well, we cluster the semantic space using k-means clustering so that the resulting number of clusters be up to $2 \times N \times K$, and then randomly select $N$ points from the centroid set as pseudo-negatives for the primitive SVM model.

Increasing number of positives is not such an easy task considering the fact that positive examples are usually not from the target set, and their distribution might be different that the target space distribution. Thus, over-sampling the data in the query semantic space might further skew the SVM learning, and strongly influence the performance. Instead, we consider the data points in the target semantic space i.e. potential near-duplicates of the positive examples on the targeted space. Probability for near-duplicate positive example is low since the examples are usually not from the targeted set. We investigate various approaches to pseudo-positive sampling of points from the dataset:

*RANDOM* Establish a low threshold $\epsilon = 0.01$ distance. If selected data point is within range of positive example, we treat it as a positive example, as increasing the number of positives will not only infuse training process.

*BAGGING* Bagging approach uses random data sampling and clusters selected data samples in order to select a set of pseudo-negatives for primitive SVM approach.

*CLUSTER* From $2 \times N \times K$ cluster centroids, for each of $P$ external examples per topic, select the cluster centroid closest to that data point, and treat it as a positive example.

*OUT* approach uses the same approach as cluster, but the pseudo-negatives are not sampled from the targeted set but from the outside set in the same domain. This approach is feasible only for visual-based approach as we use 2005 development set as an outside set.

*FUSED* approach fuses *CLUSTER* and *OUT* approach using statistical averaging.

Training is further boosted by assigning a positive label to a set of clusters closest to the positive data points which allows for the larger selection of pseudo-negatives $N$ from up to $2 \times K \times N$ cluster centroids.

## 4.4 Visual-based retrieval

ur visual-based approach is shown in Figure 6(a). Descriptor selection is a difficult tasks since we don't know the relationship of features to the semantics of individual queries. We selected top 4 diverse descriptors based on their overall most robust MAP in previous experiments [NNT05], as described in Section 2. All the approaches were tested for $K$=10, and $N$=10*$P$. Bagging method exhibited low MAP in these experiments. This is not surprising, since using pseudo-negatives from only one cluster can actually enable low selectivity in a high-dimensional feature space. The overall improvement of the cluster methods over baseline MECBR (up to 100% for the local color), and over SVM random baseline (up to 35% for texture). Next, we fuse the visual runs using proposed combination hypothesis and data modeling approaches, as shown in the Table 4.4. This experiments confirms our findings that better modeling of the input space is relevant when topic topics have low AP. 2005 dataset contained more visually relevant queries, and fusing the visual runs over descriptor spaces results in the close AP, regardless of the data modeling method. 2006 dataset contained small number of visually "simple" queries, and thus the performance measure was strongly influenced by data modeling methods, resulting in average improvement close to 500 %. We find that applying multiple biased sampling and selection method across variety of features results in enhanced performance over any of the baseline models. More importantly, we have proved that the sophisticated approach to modeling of the training samples improves the visual search and consistently improves the text baseline over range of visual samples and range of visual support of the diverse topics in TRECVID benchmark: up to 53.43 % for 2005 and 21.54 % 2006 TRECVID topics. We are working on context-based modeling of negative samples for each primitive model, and on further up-sampling positive examples.

| **Visual** | RANDOM | CLUSTER | OUT | FUSED |
|---|---|---|---|---|
| 2005 | 0.0877 | 0.0853 | **0.0882** | 0.0880 |
| 2006 | 0.0012 | 0.0040 | **0.0072** | 0.0065 |

Table 3: Data sampling influence on mean average precision (MAP) of the fused visual runs over methods and descriptors

## 4.5 Semantic-based retrieval

Semantic space is different than the low-level descriptor space. In practice, the state-of-art is to apply low-level

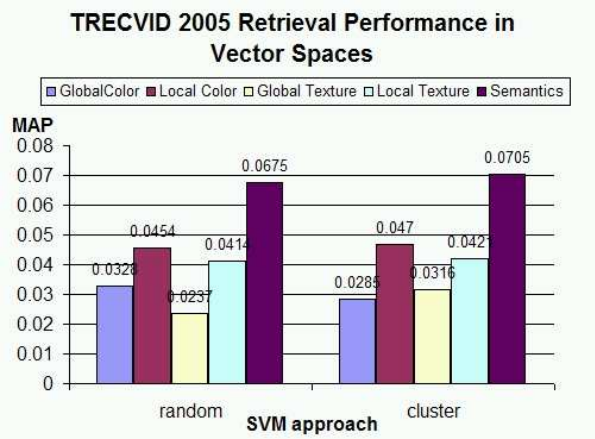



Figure 8: Retrieval performance of two svm-based data modeling approaches in the low level descriptor space (color and texture) and in the semantic space evaluated on TRECVID 2005 topics.

image feature extraction techniques to the visual data and build classifiers from the extracted features. However, feature, parameter, and method selection for each of the concepts varies, and models, in general, do not share this commonality. Thus, semantic space is highly non-linear as the dimensions it is comprised of use different approaches and parameters. Euclidean distance as a measure of closeness and distance does not make much sense in this space. Thus, we adapt our baseline method, and use only SVM portion of it, as MECBR does not make sense. As for the data modeling approach, as we do not have any development sets to learn semantic models, we compare *CLUSTER* approach to *RANDOM* baseline one. To further examine the feasibility of search in semantic spaces, we compare the data modeling results in different vector spaces for random and cluster data modeling methods over TRECVID 2005 dataset. We compare the performance in the four chosen descriptors to the performance in the 39-dimensional semantic space, as shown in Figure 8. We see that data modeling in semantics space outperforms modeling in any of the descriptors space by 50% to 180% for both approaches, and can potentially enhance content-based search.

We proposed to use cluster method as a way to compensate for over-fitting on the skewed data distribution, and to diversify the data in the modeling setup, both positive and

| **SEMANTIC** | KNN | SVM RANDOM | SVM CLUSTER |
|---|---|---|---|
| 2005 | 0.00008 | 0.06748 | **0.07055** |
| 2006 | 0.00146 | 0.03299 | **0.03698** |

Table 4: Data modeling influence on mean average precision (MAP) of the individual semantic runs

negative ones. In conclusion, more robust modeling of the semantic space results in improved baseline semantic performance of over 12% in the wide range of complex rare topics and video datasets.

## 4.6 Multimodal Fusion and Reranking

The final component of the IBM automatic search system this year was the emphasis on multimodal fusion. We have explored query-dependent search fusion among the text, model, semantic and visual retrieval scores. We tried out three different fusion approaches – a query-independent non-weighted fusion approach, and two query-class-dependent fusion approaches using strict and fuzzy query class assignments.

We analyze the input query text in order to generate query features and assign them to query classes. We use the semantic analysis [uima, CCCP$^+$04] engine to tag the query text with more than a hundred semantic tags, the tags include person, geographic entities, objects, actions, events, etc. For example, "Hu Jintao, president of the People's Republic of China" would be tagged with "Named-person, President, Geo-political Entity, Nation".

*Qclass*: query-class dependent weights. We assign each query into one of seven pre-determined classes. Ties are broken according to the concept detectors or retrieval engine performance in the state-of-the-art. Weights for each class are taken as the set that maximized the average performance metric for all training queries in the class. For non-differentiable performance metrics, this can be done by either exhaustive search on a few dimensions, or heuristic search with restart on a few dozen dimensions.

*Qcomp*: query-component dependent weights. This extends *Qclass* by allowing overlap in the seven query features. An optimal weight are similarly learned over the set of training queries with this component by maximizing the average performance metric. Weights for a new query

is computed as averaging the optimal weights among all of its active components.

We use the 24 queries from TRECVID 2005 as training queries to learn a set of linear combination weights. The per-class or per-query weights are learned with exhaustive search over the text, model, and visual scores. In strict query class assignments the new queries would use the optimal weights for that class, in fuzzy query assignments the new queries would use a mixture of the optimal query-specific weight based on the cosine distance of the new query to the training queries. *Qclass* and *Qcomp* query-dependent fusion schemes had yielded 14% and 13% relative improvement from query-independent fusion, respectively. These approaches generated our best runs with MAP scores of 0.076, 0.086, and 0.087.

## 4.7 Experiments and Results

We submitted 5 automatic type A runs for this year's Search Task, which are listed with their corresponding MAP scores in Table 5 and in Figure 9.

| Run ID | Run Description | Run MAP |
|---|---|---|
| F_A_1_JW_Base_6 | Text | 0.0405 |
| F_A_2_JW_Story_3 | Text+Stories | 0.0518 |
| F_A_2_JW_Qind_5 | Simple Fusion | 0.0756 |
| F_A_2_JW_Qcomp_2 | Strict Fusion | 0.0855 |
| F_A_2_JW_Qclass_4 | Fuzzy Fusion | 0.0867 |

Table 5: Mean Average Precision scores for all IBM automatic search submissions.

Our text-based system used JuruXML semantic search engine and several UIMA components developed by IBM Research for advanced text analytics. Baseline text run had our lowest Mean Average Precision of 0.0405, but but performed competitively as it ranked in the top 20 of all automatic and manual runs as shown in Figure 9. We also considered retrieval at the news story level, with story boundaries automatically extracted and provided by Columbia University [HC05]. This resulted in MAP of 0.0518 and 11th best overall run, see Figure 9. Our top three runs were based on the fusion with ranked lists generated by speech-based, visual-based, and semantic-based runs, and re-ranked using model-based approach. First, runs were fused using simple non-weighted averaging of

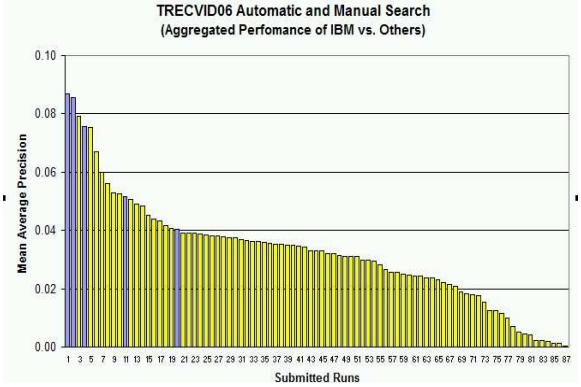



Figure 9: Mean Average Precision performance of automatic and manual submitted runs. IBM Research runs in blue, others in yellow.

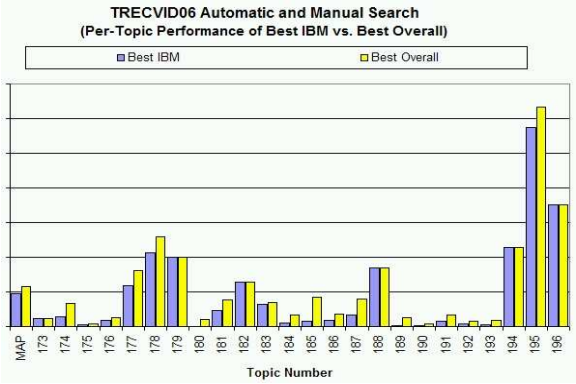



Figure 10: Average Precision comparison of the best IBM automatic search type A per-topic result vs. best overall automatic and manual type A per-topic one.

statistically normalized scores resulting in 0.0756 MAP and 87% improvement over text-only baseline.

The highlight of our system this year were the top performing two query-class-dependent fusion approaches using fuzzy and strict query class assignments. In strict query class assignments the new queries would use the optimal weights for that class, in fuzzy query assignments the new queries would use a mixture of the optimal query-specific weights. The *Qclass* and *Qcomp* query-dependent fusion schemes has yielded 14% (0.0867 MAP) and 13% (0.0855 MAP) relative improvement from

query-independent fusion and 115% and 111% improvement over text-only baseline, respectively, see Figure 9. Detailed per-topic analysis of the best overall average precision of all submitted automatic and manual type A runs vs. best IBM automatic type A run in shown in Figure 10. Mean average performance of best IBM automatic type A runs over individual 24 topic is 0.0951. Overall, our improved speech, semantic and visual approaches and query dependant fusion approaches were the key performance contributors for our system.

# 5 Interactive System Improvements

In this section, we present some of the improved capabilities of the Marvel system that allow for (a) automatic labeling and grouping of multimedia content using existing metadata and semantic concepts, and (b) interactive context driven tagging of clusters of multimedia content. Proposed system leverages existing metadata info in conjunction with automatically assigned semantic descriptors.

## 5.1 Indexing Multimedia Content

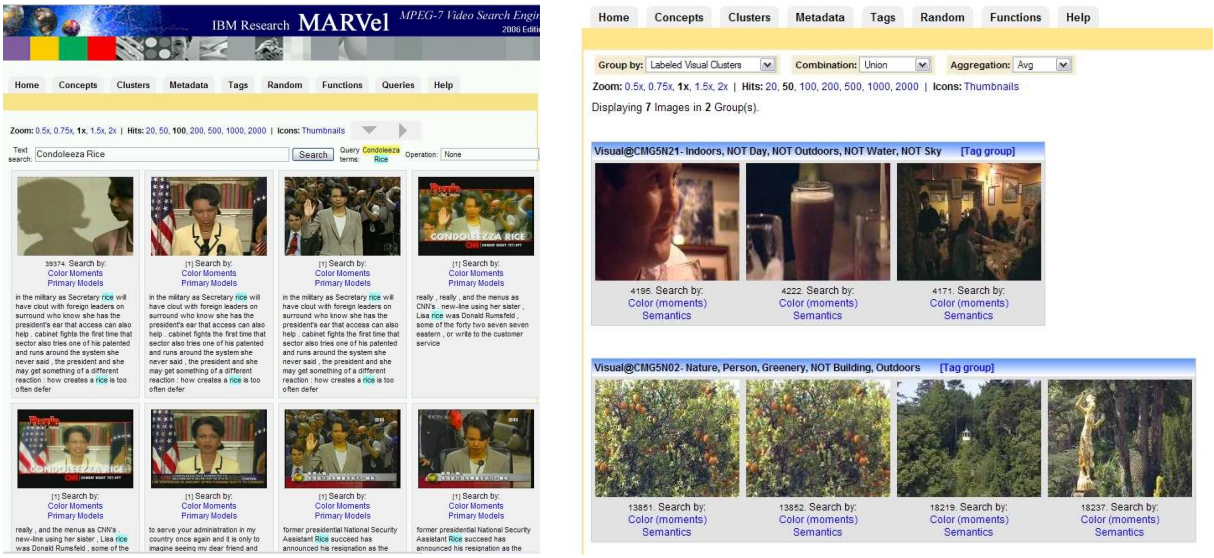
**Metadata** Digital image metadata, information about digital images, plays a crucial role in the management of digital image repositories. It enables cataloging and maintaining large image collections, and facilitates the search and discovery of relevant information. Moreover, describing a digital image with defined metadata schemes allows multiple systems with different platforms and interfaces to access and process image metadata. Importance of metadata and its widest use propelled the development of new standards for digital multimedia data schemes. These metadata schemas provide a standard format for the creation, processing, and interchange of digital multimedia metadata, and enable multimedia management, analysis, indexing, and search applications [Tes05].

**Automatically Tagged Semantics** Explicit modeling of semantics allows users to directly query the system at a higher semantic level. For example, powerful techniques have been demonstrated in the context of the NIST TRECVID video retrieval benchmark [AAC+05]. Fully-automatic approaches based on statistical modeling of low-level audio-visual features have been applied for detecting generic frequently observed semantic concepts such as indoors, outdoors, nature, man-made, faces, people, speech, music, etc. Statistical modeling requires large amounts of annotated examples for training. Since this scenarios is not feasible in the rushes archive, we adopt a new approach for automatic semantic tagging. We re-use existing semantic models, trained on the produced news and multimedia data, to automatically associate confidence scores of rushes data with those cross-domain concept models. To enable cross-domain usability, we chose the general semantic models from LSCOM [NST+06] lexicon, based on the consistent definitions of the concept across different multimedia and video domains (photo albums, web, news, blogs, raw video).

**Cluster labeling** In this demo we present a novel approach for labeling clusters in minimally annotated data archives. We propose to build on clustering by aggregating the automatically tagged semantics. We propose and compare four techniques for labeling the clusters and evaluate the performance compared to human labeled ground-truth. We define the error measures to quantify the results, and present examples of the cluster labeling results obtained on the BBC stock shots and broadcast news videos from the TRECVID-2005 video data set[TS06].

## 5.2 Overview of Interactive System Improvement

Interactive search in Marvel consisted of searching by visual features, text-based and model-based search. Although these techniques are very powerful, we want to enable the user to enrich the content with subjective interpretations of the content. Recently, we enriched our system with the functionality of *TAGGING* and *GROUPING* of video shots or images. While tagging manipulates the metadata, grouping improves the visualization of query results. Figure 12(a) shows an example flow of how a user could retrieve a meaningful result set from the system. The first step is to collect an initial set of multimedias by e.g. querying for the text ”basketball”. The resulting set is grouped by e.g. corresponding semantic clusters, generating groups with labels like ”Person, Studio, Indoors” or ”Military, Vehicle, Road”. The displayed




(a) Interactive Search Example

(b) Grouping using Clusters

Figure 11: (a) IBM Marvel multimedia analysis and retrieval system used for interactive search (results for qry194), and (b) first page search results grouped by visual clusters.

groups can be immediately tagged with whatever the user associates with them. In Figure 12(a), such a tag could be ”Jack’s birthday”.

### 5.2.1 Semantic grouping of Query results

Assume a user, having collected a sizeable video data set for the topic of interest would like to visually summarize video content before deciding on the next step. Our system offers the possibility to improve the visualization of query results by grouping them using existing metadata and clusters. Depending on which data were extracted, we can group by certain EXIF metadata [Tes05] like flash/no flash, date when the picture was taken, any metadata associated with the particular video shot (i.e. video name, channel etc.) as well as by automatically labeled visual and semantic clusters [TS06]. The groups are computed dynamically, by initiating on the result set which is currently displayed on the screen, and the following steps are taken:

1. determine the grouping category (e.g. visual clusters)
2. collect group labels for every single multimedia in the current result set that matches the selected category
3. group images/shots in the result list by common label
4. put all images/shots belonging to the same group into a visual container labeled with the group label and display them as shown in figure 11(b)

Note that the order, in which the images/shots were arranged in the original set reflects the relevance of the search result in descending order. We try to preserve this order in the groups as good as possible. Whatever group the first multimedia in the result set belongs to will always be displayed as the first group. If the second multimedia belongs to the same group, we proceed to check the next multimedia. Figure 11(b) illustrates how groups are visualized to the user. This result set was grouped by visual clusters and shows the value of the grouping feature very well. The first group contains items belonging to the visual cluster ”Indoors, NOT Day, NOT Outdoors, NOT Water, NOT Sky”, items in the second group belong to the visual cluster ”Nature, Person, Greenery, NOT Building, Outdoors”.


(a) New System Functionalities (b) Tag Cloud

Figure 12: (a) Overview of the summarization, grouping, and event tagging capabilities the interactive system, (b) a tag cloud visualizes the most frequently assigned tags.


Figure 13: Single item and associated metadata overview in IBM Marvel with the possibility to add/remove tag(s)

### 5.2.2 Tagging of multimedia content

The tagging concept has recently become very popular among internet users. People upload their personal photos to online communities, share them with other users and assign keywords (i.e. Tags) which describe the content from a personal point of view. Tags are freely chosen labels that help to improve a search engine's effectiveness because content is categorized using a familiar, accessible and shared vocabulary. The labeling process is called Tagging. The idea of assigning metadata to web pages has a long history. Since text search engines like Altavista, Google and Yahoo came up, authors of web pages used the HTML 'meta' directive to assign keywords that describe the content. Recently, this idea has also become very popular in the field of multimedia search engines. The IBM Marvel system offers several ways (low-level search, model-based search, text search) to retrieve items that match the topic of interest. The latest feature added to Marvel enables the user to assign subjective tags to multimedia content. The basic idea behind the introduction of the tagging concept is to enable event-based annotation. Assigning the same tag to different group of items can describe an event that one can search for later on. Tagging in Marvel currently covers:

- add/delete one or more tags to/from a single shot
- add/delete one or more tags to a group of shots
- add or more tags to an arbitrary query result
- search for shots that were tagged with the same label
- visualize the most frequent tags in the collection

Figure 5.2.1 shows how operations applied to single multimedias are integrated into the Marvel user interface. We use different confidence values, to distinguish such "group tags" from tags that were assigned to a single shot. Although we don't evaluate the confidences for tags, we might consider doing that in the future.

Once a reasonable number of tags has been assigned to the multimedia collection, it's meaningful to get an

overview of the most frequent tags. Therefore we implemented the commonly known idea of the ”Tag cloud” in which most frequently used tags are depicted in a larger font, while the displayed order is alphabetical (see Figure 12(b)). Showing all tags would make the tag cloud unreadable, so we only consider the top 2000 tags. When hovering over a tag within the cloud, a tool-tip appears saying how many pictures are associated with this tag. This features enables smooth browsing and simplified view of one domain when we have a high number of tags/concepts/videos/etc.

### 5.3 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. This year's improvements to the system include more user-friendly interface, extended capabilities using existing metadata, better summarization of target search data using clustering, grouping and intersection functions. 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. This year, the results from an automatic run were used to kickoff the interactive search. Figure 11(a) illustrates the Marvel multimedia analysis and retrieval system. An on-line demo of the system can be accessed from http://mp7.watson.ibm.com/marvel/. IBM Marvel interactive search run MAP was 0.1216. Detailed inspection of the results revealed that our cut-off limit was set too high. As a result, third of the dataset was not ingested in the system nor evaluated.

### 5.4 BBC Rushes

If there is no information about the multimedia content, the only effective search is to browse through the numerous folders to find the right photo or video shot. Multimedia management programs have the capability to extract knowledge from heterogeneous data sources, and to reduce the cost of annotation and labeling in an interactive environment. However, one of the challenges of these multimedia retrieval systems is to organize and present the video data in such a way that allows the user to most efficiently navigate the rich index space. The information needs of users typically span a range of semantic concepts, associated metadata, and content similarity. We propose to jointly analyze and navigate metadata, semantic and visual space for the purpose of identifying new relationships among content, and allowing user to link the aggregated content to a complex event description. As a result, intersection of different modalities, semantic grouping of search results, and tagging capability on the group level in IBM Marvel system greatly help summarize and overview the content of this year's BBC rushes dataset. Cluster labeling [TS06] helped us summarize and select relevant visual and semantic clusters in BBC rushes data. Moreover, we have the capability to tag a result grouping set that was dynamically collected using e.g. low-level feature search or model-based search to assign a high-level human interpretation to this specific result. Let's say a user retrieves all multimedias that show some kind of sports (query: Concepts@sports) and intersects it with multimedias that show soccer (low-level feature search). Having collected this set of multimedias, the user might assign tags like ”soccergame”, ”germany” and ”worldcup” or any other high-level interpretation he/she associates with the result set. This enables user to tag events discovered in rushes-type of dataset. An online demo of the BBC 2006 rushes can be accessed from http://mp7.watson.ibm.com/BBC/.

## 6 Conclusion

IBM Research team participated in the TREC Video Retrieval Track Concept Detection, Search, and Exploratory tasks. In this paper, we presented preliminary results and experiments for the Search task. More details and performance analysis on all approaches will be provided at the TRECVID06 Workshop, and in the final notebook paper.

## References


[AAC+05] Arnon Amir, Janne Argillander, Murray Campbell, Alexander Haubold, Shahram Ebadollahi, Feng Kang, Milind R. Naphade, Apostol Natsev, John R. Smith, **Jelena Tešić**, and Timo Volkmer. Ibm research trecvid-2005 video retrieval system. In *NIST TRECVID-2005 Workshop*, Gaithersburg, Maryland, November 2005.

[AKJ04] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *15th European Conference on Machine Learning (ECML)*, 2004.

[BLJ] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004)*, Baniff, Alberta, Canada, July.

[BP03] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Joint Conference on Artificial Intelligence*, pages 805–810, Mexico, Aug. 9-15 2003. Morgan K.

[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.

[EXCS06] S. Ebadollahi, L. Xie, S.-F. Chang, and J. R. Smith. Visual event detection using multidimensional concept dynamics. In *Interational Conference on Multimedia and Expo (ICME)*, Toronto, Canada, July 2006.

[HC05] Winston Hsu and Shih-Fu Chang. Visual cue cluster construction via information bottleneck principle and kernel density estimation. In *The 4th International Conference on Image and Video Retrieval (CIVR)*, Singapore, July 2005.

[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), 1999.

[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.

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

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

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

[NNT05] 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.

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

[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.

[OIKS] P. Over, T. Ianeva, W. Kraaij, and A.F. Smeaton. Trecvid 2006 an introduction. In *NIST TRECVID-2006 Workshop*.

[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.

[PBP03] S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico, Feb. 16-22 2003. Springer.

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

[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.

[Stu99] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11, 1999.

[Tes05] J. Tešić. Metadata practices for consumer photos. In *IEEE Multimedia Magazine*, volume 12, July 2005.

[TLN+03] Belle L. Tseng, Ching-Yung Lin, Milind R. Naphade, Apostol Natsev, and John R. Smith. Normalized classifier fusion for semantic visual concept detection. In *ICIP*, 2003.

[TS06] Jelena Tešić and John R. Smith. Semantic labeling of multimedia content clusters. In *International Conference on Multimedia and Expo(ICME)*, Toronto, Canada, July 2006.

[uima] UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*.

[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.