Proceedings of (VIDEO) TREC 2003 (Twelfth Text Retrieval Conference), Gaithersburg, MD, November 17-21, 2003.

Informedia at TRECVID 2003: Analyzing and Searching Broadcast News Video[1]

A. Hauptmann, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng,
N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, R. Yan, and H.D. Wactlar

1 Overview

1.1 Semantic Classifiers

We submitted a number of semantic classifiers, most of which were merely trained on keyframes. We also experimented with runs of classifiers were trained exclusively on text data and relative time within the video, while a few were trained using all available multiple modalities.

1.2 Interactive search

This year, we submitted two runs using different versions of the Informedia systems. In one run, a version identical to last year's interactive system was used by five researchers, who split up the topics between themselves. The system interface emphasizes text queries, allowing search across ASR, closed captions and OCR text. The result set can then be manipulated through:

- storyboards of images spanning across video story segments
- emphasizing matching shots to a user's query to reduce the image count to a manageable size
- resolution and layout under user control
- additional filtering provided through shot classifiers such as outdoors, and shots with people, etc.
- display of filter count and distribution to guide their use in manipulating storyboard views.

In the best-performing interactive run, for all topics a single researcher used an improved version of the system, which allowed more effective browsing and visualization of the results of text queries using a variety of filter strategies. The improvements made included a magnifying lens on the keyframe under mouse focus in the storyboard, simplified classifier filter access and use, and a browsing interface to browse the top-ranked shots according to the different classifiers. Color and texture based image search engines were also optimized for better performance.

1.3 Manual search

Our manual search runs exploited multiple retrieval agents in the dimensions of color, texture, ASR, OCR, and some of the classifiers (such as anchor, PersonX). Different schemes were explored to combine the classifiers in either a fixed weighting or a per-query weighting scheme. Negative Pseudo-relevance feedback, which we had experimented with initially in one of last year's manual runs, has been refined and applied to most of our submissions. A new approach called co-retrieval, which uses to top results from one modality to train retrieval classifiers and weights in multiple modalities was also explored, although it did not provide the overall best result. The contrast between the different submitted runs sheds light on the potential of several of the approaches. It was surprising to us that even our text-based baseline using the OKAPI retrieval formula performed better many other runs. This may indicate the importance of manual keyword selection for queries. One implication for future comparisons is that manual keyword expansion introduces to many variabilities in retrieval performance, making comparisons across groups difficult.

2 Extracted Features and Non-TRECVID Metadata Classifiers for Anchors and Commercials

Underlying all classifiers and retrieval systems is a set of features extracted from the MPEG videos. In addition to the features described here, we also took advantage of the commonly provided speech

[1] This work was supported in part by the Advanced Research and Development Activity (ARDA) under contract number MDA904-02-C-0451 and by the National Science Foundation under Cooperative Agreement No. IRI-9817496.

information and OCR information for the FSD and FST sets. In addition to the TRECVID evaluated classifiers, we also built classifiers for anchors and commercials.

Audio Features: For the TREC2003 video retrieval task in addition to the Speech Recognition engine, a set of features that directly characterize audio content without any language modeling were developed. These features are used to assist the extraction of the following medium-level audio-based features: music, male speech, female speech, and noise. In addition they were combined with visual and text-based features for various detectors. A set of low level numerical audio features is calculated every 20 milliseconds. They are all based on the magnitude spectrum calculated using a Short Time Fourier Transform. They consist of features that summarize the overall spectral characteristics such as Spectral Centroid, Rolloff, Relative Subband energies as well as the Mel Frequency Cepstral Coefficients which are perceptually motivated features used in Speech Recognition. In addition for male/female speech discrimintation the pitch of the signal was calculated using an Average Magnitude Difference Function (AMDF). In order to capture the texture properties the means and variances of the features over a 2 second sliding texture window are computed. A multidimensional Gaussian classifier with a full covariance matrix was trained using labeled samples. The classifier makes a decision every 20 milliseconds but it uses information of the past 2 seconds. For audio-only based classification the results are returned for variable length shots. The classification accuracy is calculated as the largest class percentage of 20millisecond frames. The training was done using the IBM annotations with significant quality control to ensure that the samples used for training are appropriate.

Low-level Image Features: The color feature is the mean and variance of each color channels in HSV (Hue-Saturation-Value) color space in a 5*5 image tessellation. The hue is quantized into 16 bins. Both saturation and value are quantized into 6 bins. The texture features are obtained from the convolution of the image pixels with 6 Gabor wavelet filters. We use the mean values of twelve oriented energy filters aligned in 30-degree intervals. We compute a histogram for each filter in a 3*3 image tessellation, which is quantized into 16 bins. Their mean and variance are treated as the texture features. Another low-level feature is the canny edge direction histogram. A Canny edge detector was applied to extract the edges from images also from a 3x3 grid structure. The edge histogram includes a total of 73 bins. First 72 bins represents the edge directions quantized at 5 degree interval and the last bin represents a count of the number of pixels that are not contributed to any edges. As a preprocessing step for all these features, each dimension of the feature vectors is normalized by its own variance.

Face Features: Schneiderman's face detector algorithm [30] was used to extract frontal faces. Size and position of the largest face are used as additional face features.

Text-based features are the most reliable high-level features applicable in video retrieval, based on the best-performing video retrieval systems in 2001 and 2002. Three types of text features are processed in our system, i.e. automatic speech transcripts (ASR) which was provided by the LIMSI speech recognizer, Video Optical Character Recognition (VOCR) extracts the overlaid text. Speech transcripts are important supplementary sources to closed captions, especially in commercial footage where closed captions are not available. Closed caption is synchronized using the output of speech transcripts with corresponding time alignment information. Words in from a stopword list [27] were removed, and Porter stemming algorithm [25] was used to remove morphological variants.

Video OCR (VOCR): How can errorful VOCR data be useful? We developed a restricted approximate match technique to search for VOCR words similar to query words. The technique was built upon Manber and Wu's approximate string matching technique, which allows certain predefined number of edits (insertions, deletions, and substitutions) to transform a string to another string [21]. In an errorful text database, the approximate match will have a high probability to retrieve irrelevant text from a pool of noisy data. For example, the word "Clinton" may retrieve "Cllnton", "Ciintonfi", "Cltnton", and "C1inton", which are correct to the query word; however, it may match incorrect text like "Arlington", "EIICKINSON" (for "DICKINSON"), and "Cincintoli" (for "Cincinnati"). From empirical experience, we restricted the edit distance based on the length of query words: words with fewer characters must have a smaller edit distance.

In order to aid the search task, we associate visual context to each retrieved VOCR word for searchers to quickly determine its relevancy. We implemented the restricted approximate match associated with visual context as a side tool to suggest the use of errorful but relevant query words for the search task.

2.1 *Fisher's Linear Discriminant for Anchors and Commercials*

We used Fisher's Linear Discriminant (FLD) to develop 2 additional classifiers, for anchors and commercials. The basic idea of our multimodal combination approach is to apply FLD to every feature set and synthesized new feature vectors. For example, consider 3 different feature sets, $A:\{a_1, a_2 \ldots.., a_{n1}\}$, $B:\{b_1, b_2 \ldots.., b_{n2}\}$, $C:\{c_1, c_2 \ldots.., c_{n3}\}$, where n1, n2, n3 are the dimensions of the respective feature sets. We first apply FLD individually to every feature set. We now pick the top N eigenvectors as axes and project the feature vectors into this new space. The new feature sets will be $A_{fld}\{a'_1, a'_2 \ldots.., a'_N\}$, $B_{fld}\{b'_1, b'_2 \ldots.., b'_N\}$, and $C_{fld}\{c'_1, c'_2 \ldots.., c'_N\}$. The new vectors are not only selected from the raw data, but also generated by a discriminant function. Using these synthesized feature vectors to represent the content, we can apply standard feature vector classification approaches.

For our anchor and commercial detector evaluation we only used the sample set of the TRECVID 2003 ABC broadcast news corpus containing13 30-minutes broadcast news shows (6.5 hours). We built two different SVM-based classifiers, one is an anchor classifier, which distinguishes shots that contain an anchor from those that don't, and the other is a commercial classifier which distinguishes commercials. For the anchor classifier, we used the color histogram, face information (size, position, confidence from face detector) and speaker information as our feature sets. For the commercial classifier, we utilize color histograms and audio features. The results of applying FLD to analyze the weight of each dimension on color histogram is shown in Figure 1. From the corresponding grid weights, we discover that FLD really emphasizes the studio background. Table 1 shows the result of an anchor and commercial classifiers using different feature sets and combination strategies.

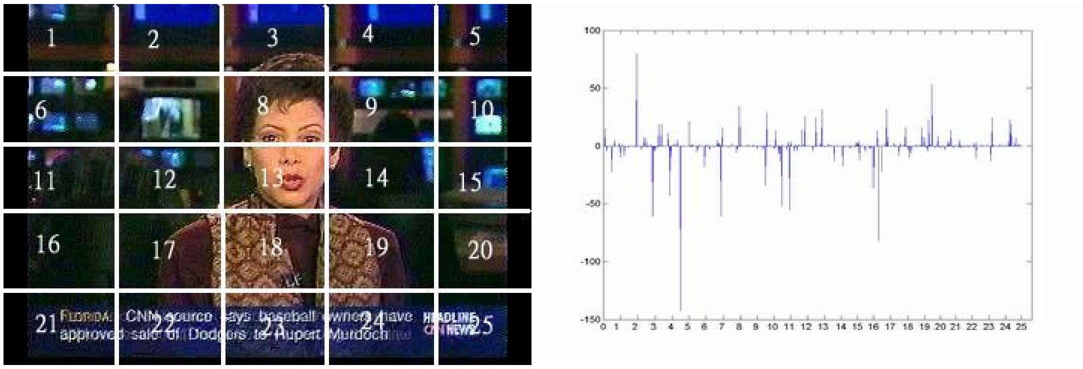

Figure 1: FLD weights for anchor detection. The the image grids or on the left, the right shows the weight of every grid. Note that grids 2, 5, 17, and 20 have high weights, indicative for studio background.

Category	Mean Average Precision Anchor	Mean Average Precision Commercial
Image	0.59	0.559
Face Information	0.47	N.A.
Speaker ID/Audio	0.6	0.737
Feature Synthesis	0.49	0.713
Meta-classifier	0.65	0.834
FLD + Feature Synthesis	0.69	0.861

Table 1: Anchor and Commercial classifier result. Image features are based on 5by5 125 bin color histograms. Face information combines size, position and confidence. Speaker ID is from the LIMSI provided data used only for anchor detection. Audio feature is the Short Time Fourier Transform used only for commercial detection. Feature synthesis simply combines all feature sets. Meta-classifier classifies all three feature sets individually first and then builds a final classifier based on those three classifiers. FLD+feature synthesis is the approach described above.

3 Feature Classifiers

3.1 Baseline SVM Classifier with Common Annotation Data

Based on the 2002 TRECVID work, we used support vector machine learning, with the power=2 polynomial as the kernel function for our baseline classifier runs. Due to the temporal correlation between adjacent images in a video, an initial cross validation based on random sampling of shots gave much better performance than appropriate for the true prediction capability of the models. This was due to the fact that similar shots appeared throughout a single video or 'movie', so we performed a video based cross validation with portions of the common annotation data. This baseline system, which used only image features (no faces) described in section 2, achieved an MAP of 0.112 for outdoors, 0.071 for buildings, 0.028 for roads, 0.122 for vegetation, 0.017 for animals, 0.040 for cars/trucks/buses, 0.059 for aircraft, 0.051 for sports, 0.017 for weather news, and 0.012 for physical violence.

3.2 Building Detection

We explored a classifier by adapting man-made structure detection method by Kumar and Hebert [19]. This method produces binary detection outputs for each of 22x16 grids. We used color and texture features together with the features extracted from the man-made structure detection results. We extracted 5 features from the binary detection outputs:

- Number of positive grids
- Area of the bounding box that includes all the positive grids
- x and y coordinates of the center of mass of the bounding box.
- Ratio of the width of the bounding box to the height of the bounding box
- Compactness, which is the ratio of the number of positive grids to the area of the bounding box.

462 images were used as positive examples, and 495 images were used as negative examples. Negative examples are chosen from the set of positive detection results when man-made structured detection method is used by itself. After FLD, a SVM classifier was built. In the official classifier evaluations, this building detector did not outperform our baseline image based building detector also trained on the common labeled data. With a MAP 0.042 (man-made structures) vs. 0.071 (baseline SVM).

3.3 Plane Detection using additional still image data

In order to build a classifier for planes 3368 plane examples were selected from web, Corel data set and from the University of Oxford data set as positive examples. Image features as described in Section 2 were used. Negative examples were selected by a nearest neighbor search that finds the most similar images from the FSD set (both CNN and ABC) to the selected positive plane images using the selected features. 3516 negative examples are used. After FLD, SVM was applied for training. Unfortunately, this detector (MAP 0.008) performed worse than our baseline detector (MAP 0.059), built only from the common labeled data set. We surmise that the additional still image training data, since it came from outside the collection, was sufficiently different in its characteristics to introduce an unwanted bias into the classifier.

3.4 Car detection

Car detection was performed with a modified version of the Schneiderman face detector algorithm [30]. Trained on numerous examples of side views of cars only (no trucks or buses), it outperformed our baseline classifier with an MAP of 0.114 vs. the baseline MAP of 0.040.

3.5 Zoom Detection

The zoom detector followed the algorithm described in [16], it performed well with an MAP of 0.632. The approach uses MPEG motion vectors to estimate the probability of a zoom pattern.

3.6 Female Speech

The female speech detector used an SVM trained on the LIMSI provided speech features, together with the face characteristics described in section 2 using the common annotation as truth. It performed well with an MAP of 0.465.

3.7 Text and Timing for Weather News, Outdoors, Sporting Event, Physical Violence and Person X Classifiers

We explored extracting features such as person x and people just based on text information, and found them to perform better than random baselines on the development data, as shown in Table 2, using only text information, plus timing if there was a strong temporal structure. Timing information is the implicit temporal structure of the broadcast news, especially weather reports and sports news. The density function can be estimated using kernel density estimation with Gaussian kernel and Sheather-Jones bandwidth selection method, plotted in Figure 2. While weather news in ABC is more likely to appear early in the program, weather news in CNN appears at a specific time. Sports news is usually shown late in the ABC program, and again, the timing cues for sports news in CNN news are stronger.

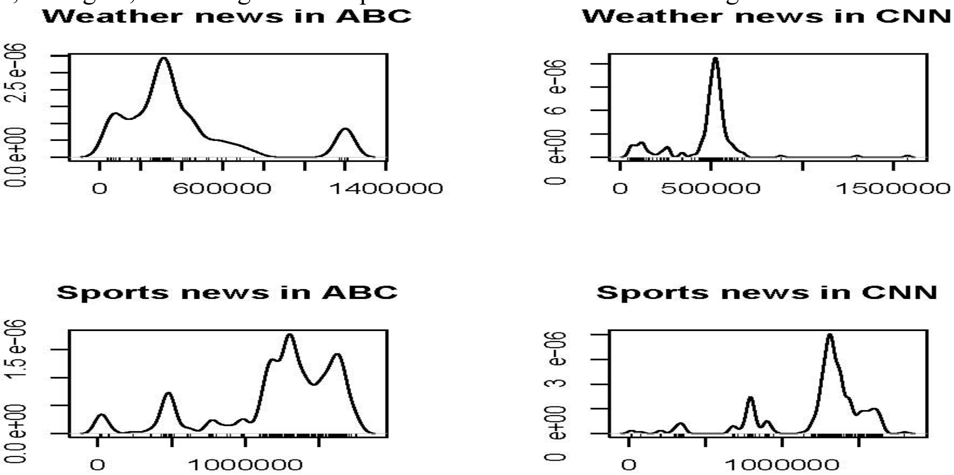

Figure 2: Non-parametric estimations of probability density functions for weather and sports news in ABC and CNN. The X-axis is the average of the starting and ending time of the shots in milliseconds.

Task	Channel	MAP	Random Baseline
Weather News	CNN	0.7888	0.0130
	ABC	0.1176	0.0056
Sporting Event	CNN	0.0614	0.0433
	ABC	0.0398	0.0146
Building	CNN	0.0614	0.0433
	ABC	0.0603	0.0484
Road	CNN	0.0300	0.0217
	ABC	0.0432	0.0313
Animal	CNN	0.0548	0.0130
	ABC	0.0202	0.0103
Car/Truck/Bus	CNN	0.0749	0.0429
	ABC	0.0900	0.0551
Aircraft	CNN	0.0200	0.0093
	ABC	0.0571	0.0091
Physical Violence	CNN	0.0050	0.0038
	ABC	0.0049	0.0048
Person X	CNN	0.0868	0.0038
	ABC	0.2607	0.0041
Outdoor	CNN	0.0931	0.0718
	ABC	0.1031	0.0943
People	CNN	0.1743	0.1228
ABC	ABC	0.1706	0.1282
	C-SPAN	0.4767	0.2856
Vegetation CNN		0.0072	0.0079
ABC		0.0174	0.0161

Table 2: The results of feature extraction on the development set based only on text classifiers

All classifiers use only labels from the common annotations and data in the development set only, all runs of the submission should belong to category A in the NIST definition, although they were labeled as category C due to the constraint that every feature detection result in one run must of the same category.

For each news station (CNN, ABC, CSPAN) in the TRECVID03 corpus, one SVM [35] model was trained on the data only from that station. In order to combine results from three sources the output value of the SVM was transformed into a posterior probability, i.e. $\Pr(yi|xi)$ using logistic regression [24]. The prediction from the three sources are merged and sorted by their probabilities.

The timing-based classifier is based on the estimated conditional probability $\Pr(ti|yi = +1)$ described above, where ti is the time that shot appears in the program. The conditional probability of negative examples, i.e. $\Pr(ti|yi = -1)$ is assumed to be a uniform distribution. By integrating the area under the density from the starting time to the ending time of the shot, the class-probability of the shot can be easily obtained.

For each shot, both predictions from text-based classifier and timing-based classifiers have to be considered together in order to make a combined prediction. Stacking [37] with SVM was chosen as a meta-classifier. The results of five feature extraction tasks are tabulated in Table 3. Except for weather news, the results suggest that text information of the broadcast news in the shot may not be enough to detect these high-level features. The results also suggest that the timing information can be effective when it exists in the news program, while the combination of text and timing information appears to lie somewhere in between.

Classifier Task	**Feature Set**	**MAP**	**Rank**	**Best non-CMU MAP**
Outdoors	Text	0.027	29/35	0.227
Sporting Event	Text	0.058	27/36	0.708
	Timing	0.211	15/36	0.708
	Text + Timing	0.074	25/36	0.708
Weather News	Text	0.855	2/38	0.856
	Timing	0.795	12/38	0.856
	Text + Timing	0.804	11/38	0.856
Physical Violence	Text	0.029	16/30	0.086
PersonX	Text	0.280	9/35	0.343

Table 3: Results of five feature extraction tasks

4 News Subject Monologues

A detailed description of the news subject monologue classification can be found in [31]. Based on observed inconsistencies in the common annotations we labeled about 29 hours from the training set ourselves. The face features described in section 2 were used, based on [30]. Based on the LIMSI speech annotations [12] we developed a voice over detector and a frequent speaker detector. Voice-over detection identified of speech segments as a voice over when they contained more than 1 cut for a continuous speaker. Frequent speaker camera shots are detected as the 3 most frequent speakers in a news broadcast. Video Optical Character Recognition (OCR) [28] was applied to extract overlaid text in the hope of finding people names (Figure 4).

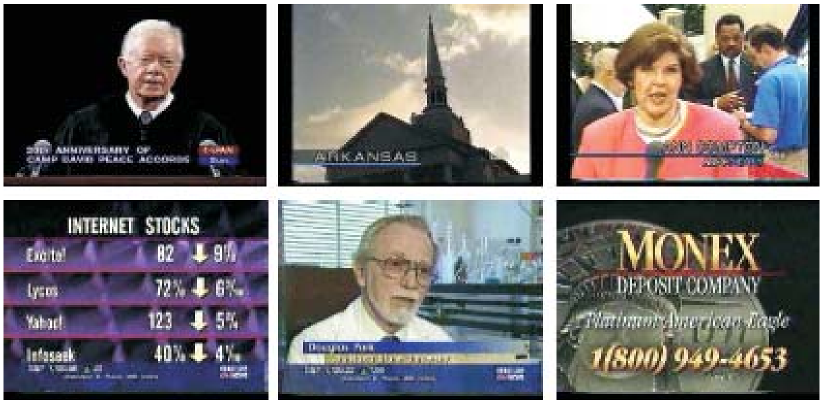

Figure 4: Common usage of overlaid text in the TRECVID corpus. From top left to bottom right: topic annotation, location annotation, reporter annotation, financial data, news subject monologue annotation, and commercial messages.

We used the total length of the overlaid text strings as a feature. Video OCR was also used as input for a named entity recognizer that is part of the Informedia system [38]. Furthermore, the detected strings were compared, using fuzzy string matching (see section 2), with a database of names of CNN and ABC affiliates. Further, we introduced a simple camera shot length feature. Another feature measures the average amount of motion in a camera shot, based on frame difference [32]. The output of the LIMSI speech recognition system [12] was also compared with a set of keywords that was found to have a correlation with reporters, financial news, and commercials. We also used commercial and anchor detectors [see section 2, above]. We combine our individual detectors and features by exploiting two well-known classifier combination schemes, namely stacked SVMs [37] and bagging [6]. For the aggregation the sum rule was used [18]. The results can be summarized with an MAP of 0.234 for training on the common annotation data, while our best system had a MAP of 0.616. Note the drop in AP when bad ground truth from the inconsistently labeled common annotation is used.

5 Finding Person X in Broadcast News

Our approach to find a specific person uses text information from a transcript and face information. Most simply, a text retrieval system can give us a first clue where the person x is mentioned, and (perhaps) visible. Using the anchor detector described above, we eliminate anchors. We simply use one specific person, “Madeleine Albright”, to explore the relationship between text information and face information. Figure 5 shows the distribution between time and the name reported in the news of the FSD set.

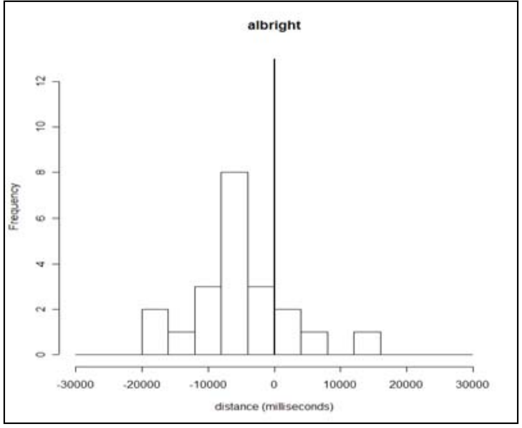

Figure 5: Relationship between the name of person x and time. The bold line shows where the name is mentioned in the transcript. The columns denote the frequency that the person is visible in the video at a given time distance. A negative distance means the person is visible after their name is mentioned.

After we find the possible positions from text information which may contain the person, we apply the distribution we trained from the data to estimate the possible shots of the person.

$$p_{name}(S) = \sigma(\mathrm{T_s} - T_0) \qquad \text{where } \sigma(\mathrm{t}) \text{ is the distribution in Figure 5} \qquad (1)$$

where S denotes one shot, T_s denotes key frame time and T_0 denotes the time of person name. However, this distribution only shows a global view. We use the anchor detector (see Section 2) to revise the distribution considering the local conditions. We used a linear combination to construct a revised distribution:

$$p_{text}(S) = \alpha p_{name}(S) - \beta p_{anchor}(S) \qquad (2)$$

where $\mathrm{p}_{\mathrm{anchor}}(S)$ is the probability that the shot contains an anchor. α and β denote the parameters to combine anchor and name distribution; p_{text} is the text information we retrieved from the transcript.

Face recognition (i.e. Eigenfaces [40]) may give us an exact match, but it is unreliable. Our approach tries to build more limited face recognition for a specific person based on video shots. We collect sample faces { F_1, F_2, F_3 ….., F_n } for person x and all faces { f_1, f_2, f_3 ….., f_m } of i-frames in the news shot which P_{text} is larger than zero. We build the eigenspace for those faces { f_1, f_2, f_3 ….., f_m, F_1, F_2, F_3 ….., F_n } and represent them by the eigenfaces { $eigf_1$, $eigf_2$, $eigf_3$ ….., $eigf_m$, $eigF_1$, $eigF_2$, $eigF_3$ ….., $eigF_n$}. For each face ($eigf_d$), there are n rank numbers based on every sample face and we combine all the results.

$$R(eigf_i) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{r_j(eigf_i)} \qquad (3)$$

where R(eigf$_i$) denotes the combination rank score for eigenface f$_i$ and r$_j$(eigf$_i$) denotes the rank of eigenface f$_i$ based on sample face F$_j$.

This combination rank gives us a similarity list of person x. We still have to estimate which shot has high possibility to contain that face.

$$S_{face}(S) = \frac{1}{k}\sum R\left(eigf_j \subset S\right) \qquad (4)$$

where k is the number of faces in shot S. S$_{face}$ denotes the likelihood that shot S contains person x's face.

Our strategy treats text and face information as features of person x in the news and builds classifiers to learn from those two features. Using "Madeleine Albright" as person x, we also obtained 20 faces from a Google image search as sample query faces. The results are shown on Table 4. First, we only used text with the distribution from Figure 5. Then, we used the result from anchor detection with text. Eigenface recognition is then applied, and also with the shot combination schema described above. Finally, we combine face information and text information together.

Category	Mean Average Precision
Text only	0.2607
Text + Anchor	0.3312
Face (i-frame)	0.1352
Face (shot)	0.1762
All	0.3791

Table 4 : The result of finding person x, for "Madeleine Albright"

6 Learning Combination Weights in Manual Retrieval

Generally, the task of multimedia retrieval, more precisely, the task of shot-based video retrieval, can be decomposed into following steps: (1) A set of features is extracted such as speech transcripts, audio, camera motion, visual features and even some semantic-level features such as anchors etc.; (2) Each shot is associated with a vector of individual retrieval scores from different media search modules; (3) Finally these retrieval scores are fused into a final ordered list via some aggregation algorithm.

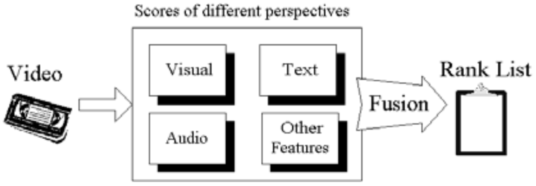

Figure 6 Fusion from different modalities

However, even if all retrieval scores are consistently useful (i.e. better than random), combination with multiple scores can not always improve on the performance of the best uni-modal score. Taking the efficiency into consideration, we adopt the weighted Borda fuse model as the basic combination approach for multiple search modules, i.e. for each shot its final score is $y=\sum_{i=1}^{n} w_i s_i$, where s_i is normalized rank-based scores, w_i are the corresponding linear weights, and *n* is the number of features. Our work last year has shown that if the linear weights w_i could be assigned appropriately, retrieval performance can be significantly boosted over that ranked by the single best retrieval score. Therefore in this framework, the retrieval task is naturally formulated as one that finds the best linear weights for various single queries. We propose two types of weight learning methods: learning weights given training labeled data and a co-training style weight learner without training data, and compared them to a baseline weighting scheme which only uses query type information.

Similarity measures: For each video frames, a harmonic mean of the Euclidean distances from each query images (color, texture, edge) is computed to be the distance between query and video frames. For retrieval from text, CC and OCR transcripts is done using the OKAPI BM-25 formula.

6.1 Negative Pseudo-Relevance Feedback (NPRF)

In TREC02 search task, we demonstrated the negative pseudo-relevance feedback score to be effective at providing a more adaptive similarity measure for image retrieval, to improve the overall retrieval performance. For the TREC03 manual search, we generated a modified version of NPRF score, which follows the idea of original NPRF algorithm, however, with modification to negative example sampling.

In the original NPRF algorithm, we choose the unlabeled data farthest from positive data as the negative sample, which is also a common strategy used in some earlier work of positive-based learning or self-learning. But if this strategy works, an underlying assumption is made that positive data are more likely to be in the boundary of the data set. It is obvious that the farthest points from the positive data are not necessarily an accurate sample for the negative data. However, a common case is that data points are clustered in several separate "clouds", which is obviously a bad case if only the farthest negative data is sampled. Therefore, we propose a better strategy to sample negative examples. This idea is inspired by the Maximal Marginal Relevance (MMR) criterion proposed by Carbonell et al [41], which takes both "relevance" and "novelty" into account. Similarly, we propose a Maximal Marginal Irrelevance (MMIR) criterion, which can account for "irrelevance" and "novelty" simultaneously when sampling the negative data. A data point has high marginal irrelevance if it is likely to be negative data and dissimilar to previous selected negative data. Formally, the MMIR criterion can be written as,

$$MMIR = arg \min_{D_i \in T \setminus S} \left\lfloor \lambda Sim_1(D_i, Q) + (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \right\rfloor$$

where *T* is the collection, *S* are selected negative images, *Q* are query images and D_{ij} are collection images.

6.2 The Value of Intermediate-Level Detectors

It is almost always the case that users want to explicitly exclude the video shots that contain commercials or anchor scenes. We apply both anchors detection results and commercials detection results to filter out useless scenes in retrieval results (see section 2).

Against the characteristics of text-based features, we observed that low-level visual feature and intermediate level detector features could provide a reasonable ranking on how closely the video shots are related to the given examples in specific domains. For example, in the query “Finding a Dow Jones graph” or “Finding a baseball player”, low-level color feature can rank the images examples very well, because the query can be represented in terms of visual information. But none of these features can capture the semantic meaning of video shots and therefore the retrieval results based on these features alone are mostly dissatisfied. Our experience indicates that text-based feature is good at global ranking and other features would be useful in refining the ranking afterwards.

6.3 Learning Weights for each Modality in Video Retrieval

Baseline: Setting weights based on query types. We roughly grouped the 25 queries into two types: queries on finding persons and other non-person queries. There are 5 queries on finding persons in all 25 queries. For these queries, the linear weights are set to be *w = (text 2, face 1, color 1, anchor 0)*. For most of the non-person queries, the linear weights are set to be *w = (text 2, face -1, color 1, anchor -1),* except when querying on aircraft and animal the linear weights is *w = (text 2, face -1, edge 1, anchor -1)* because edge features are found to be superior to color features when we develop the corresponding detectors.

Learning weights using training labeled set. Typically, supervised learning algorithms require a number of labeled training data as input. In order to perform weight-learning algorithm directly, for each query topics we collect a set of truth video shots in the development set using Informedia clients. This labeled ground truth can be fed into any kind of learning algorithms to learn the weights. However, the learning objective is somewhat different from the canonical classification framework, which is to minimize the classification error, instead we want to maximize mean average precision [42].

Co-Retrieval. As mentioned before, a straightforward approach to tune the weights is to collect a set of ground truth/training data in a training pool of video and then use them as an input of a learning algorithm to tune the weights. However, it is generally implausible for a normal user to collect enough training data on the fly. Alternative approaches have to be developed to generate reasonable weight assignment without acquiring large amount of human effort to collect training data. An idea, which naturally comes to mind, is related to the work called multi-modality learning [4]. This work basically assumes an important characteristics that sensory data is coherent across times and across sensory channels. They explore the correlation between multiple modalities information and suggest minimizing the disagreement between outputs or similarly, maximizing the mutual information between modalities [3].

Another axis of research is the co-training algorithm, where the features in the problem domain are naturally divided into two disjoint sets (or two modalities) and, under the conditional independence and sufficient learning assumption, a PAC-style bound on learning from labeled and unlabeled data holds. The essence of co-training algorithm is to learn a noise-tolerant learning algorithm A2 using the noisy labels provided by another learning algorithm A1. [23] showed that an independent and redundant feature split is quite important for the performance of a co-training algorithm. It is interesting to explore how the relation between modalities can be used to learn the their weights, and co-training is exactly one of the algorithms that we need. However, it is not feasible to directly apply co-training to multimedia data, because many features alone are not sufficient to learn the concepts, especially visual features from color or texture.

Exploiting the observations on various features, we describe a multimedia retrieval algorithm that automatically provides the weights for different modalities based on the principle of modalities coherency. The system used a variant of co-training, called co-retrieval, to exploit unlabeled data. Specifically, a set of video shots are first labeled as relevant shots using text-based features, and the results are augmented by learning with the other visual and intermediate level features. Their combination is supposed to find the video shots close to given examples, but which also largely agree with the original text retrieval.

In order to apply similar idea to co-training, we manually separated the retrieval scores into two groups,

$$y = \sum_{i=1}^{n_1} w_{1i} s_{1i} + \sum_{j=1}^{n_2} w_{2j} s_{2j}$$

where $n_1 + n_2 = n$. The feature set $(s_{11}, ..., s_{1n_1})$ and $(s_{21}, ..., s_{2n_2})$ are two sets of media features which are conditional independent to each other. The target function is the linear combination function. Based on our observation before, we typically set $(s_{11}, ..., s_{1n_1})$ to be the retrieval scores from text-based features as semantic features, and $(s_{21}, ..., s_{2n_2})$ to be retrieval scores from the other features such color histogram or face detector as non-semantic features. Iteration learning (learning weights iteratively) in co-training is not suitable in this case, because the prediction powers of these two groups are not identical. In this setting, we would like to use first group of features to generate an approximated / noisy labels *f'* for every video shots ***D***. Then, train the combination $\sum_{j=1}^{n_2} w_j^2 s_j^2$ from the labels *f'*. The entire algorithm are described as follows,

1. Grouping: Break the feature set into two groups. Typically, the first group is based on text features and the second group comprises all other features;
2. Score Generation and Labeling: Generate the retrieval scores s from different features. Label some pseudo-positive data based on first group of scores.
3. Learning: Use the pseudo-positive data to learn the linear weights
4. Final score generation: Combine the scores with the learned weights.

For the score generation step, users are required to manually define the first group of features, which they believed best fit the concept. As for the learning stage, there are a good number of linear classifiers available in the literature such as perceptrons, logistic regression and SVM. We choose the logistic regression to learn the weights, i.e. given the label *l*, maximize the logistic regression loss function,

$$-\sum_{i=1}^{m} \log(1 + \exp(-l_i \sum_{j=1}^{n} w_j s_{ij}))$$

From another point of view, this is also minimizing the cross entropy / mutual information between two different labels. A regularization factor could be added to avoid overfitting.

6.4 Experimental Results

We submitted a total of 7 manual search runs to TRECVID this year. The details of each run is listed as follows, the actual text portion of the queries is shown in the Appendix.

- CMU-PX: Most queries are the same CMU-FSD, except 5 Person X queries (query 103, 114, 118, 119, 124) treated as described in section 3. System uses NPRF
- CMU-FSD: Learning linear weights of different features based on development set ground truth. 400 shots are returned from text retrieval, using NPRF.

- CMU-400: Setting weights based on query types. 400 shots are returned from text retrieval. No NPRF
- CMU-1000: Learning linear weights of different features with development set ground truth. 1000 shots are returned from text retrieval, using NPRF
- CMU-CoRet: Co-Retrieval using top 100 shots as PRF truth to learn weights, using NPRF
- CMU1: similar to CMU-400 but with 200 shots only, no NPRF
- CMUBase: audio only (text retrieval [ASR, CC + OCR] CMU-400) TF/IDF Okapi, keyword expansion from FSD set

Experimental results of overall mean average precision are plotted in Figure 6. The experimental results indicate noticeable performance improvement can be achieved for some queries when weights are automatically learned. However, it is not clear how to provide consistently better retrieval result than a baseline performance. Part of the problem is that the training data is either too small or not representative.

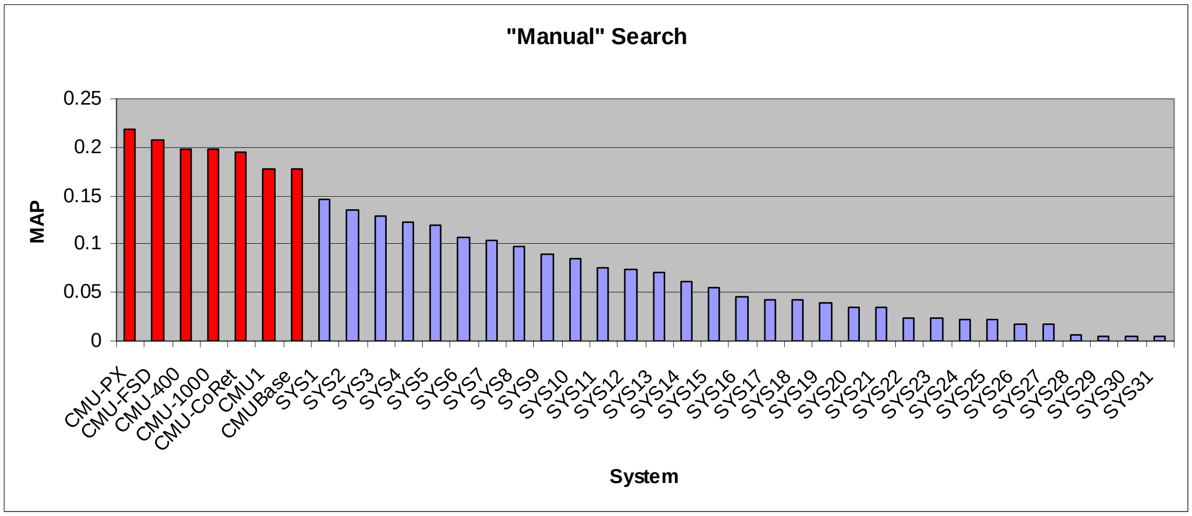

Figure 6. Overall MAP of CMU's 7 manual runs against other systems

7 Interactive TREC Video Retrieval Evaluation for 2003

A concentrated effort was made by the authors to develop an interface allowing a human to succeed with video topics as defined in TRECVID 2001. This interface was part of the TRECVID 2002 interactive query task, in which a person could issue multiple queries and refinements to the video corpus in formulating the shot answer set for the topic at hand. The interface was designed to present a visually rich set of thumbnail images to the user, tailored for expert control over the number, scale, and attributes of the images. Armed with this interface, an expert user completely familiar with the retrieval system and its features, but having no a priori knowledge of the TRECVID 2002 search test corpus, performed well on the search tasks, based on the following features [10]:

- storyboards of images spanning across video story segments
- emphasizing matching shots to a user's query to reduce the image count
- resolution and layout under user control
- additional filtering provided through shot classifiers such as outdoors, and people
- display of filter count and distribution to guide manipulation of storyboard views.

This exact system as used in the TRECVID 2002 interactive query task was again used for the TRECVID 2003 evaluation. To facilitate better visual browsing, we extended the storyboard idea to show keyframes across multiple video documents, where a "document" is automatically derived by segmenting a video production into story units through speech, silence, black frames, and other heuristics [38]. The hierarchy of information units for video is the frame, shot, document, and full production.

A set of documents is returned by a query. The shots for these documents are presented in a single storyboard, i.e., an ordered set of keyframes presented simultaneously on the computer screen, one keyframe per shot. Without further filtering, most queries would overwhelm the user with too many images. Through the use of query context, the cardinality of the image set can be greatly reduced. The search engine for text queries makes use of the Okapi method. Coupled with the user's expressed information need in the query, that need can be pinpointed to particular points in the narrative, as well as to superimposed video text. The multiple document storyboard can be set to show only the shots containing matching words. This strategy of selecting a single thumbnail image to represent a video document based on query context resulted in more efficient information retrieval with greater user satisfaction in past studies [38]. Here, the idea is extended to collapse a set of thumbnails spanning multiple documents to a smaller set of only the shots containing matches to a given query. Matches are noted on the storyboard with a color-coded marker.

To further reduce the cardinality of the storyboard, a filter can be applied based on the TRECVID shot classifiers developed at Carnegie Mellon University. For TRECVID 2003, these classifiers were indoor, cityscape, people, faces, text, news subject faces, automobiles, roads, sports, and commercials.

Pre-computed Pair-wise Shot Similarity: We pre-computed for each shot in the TREC 2003 testing set (around 80,000 shots in total) the 50 most similar shots based on the transcript. For each shot an XML file is generated that contains the information of its 50 similar shots, such as various classification labels. Thus, totally some 80,000 files are generated. These files speed up the computation in the relevance feedback process by saving the time for retrieving the shot features from the database and computing the similarity between shots. All the XML files are grouped by source (CNN, ABC, C-SPAN).

Aggregation across Shots or Videos: We generated two types of aggregation files: aggregation across shots and aggregation across video. The former type of files aggregates the consecutive shots within a certain time window in a movie. The shot based aggregation files used two time windows (1 minute and 3 minutes). The video-based aggregation file puts together the shots at a certain time window (1 minute) across 10 days' broadcast news video. For example, an aggregation file may have the shots within the 7th minute of the daily CNN news from May 1st through May 10th. Both types of aggregation files allow users to "look around" when they find an interesting shots, since neighboring shots might be interesting as well.

A new interface was developed for 2003 based on a user study with the TRECVID 2002 interface. The improvements made included a magnifying lens on the keyframe under mouse focus in the storyboard, simplified classifier filter access and use, and a browsing interface to browse the top-ranked shots according to the different classifiers. Color and texture based image search engines were also optimized for better performance. This "new" interface was evaluated as part of the interactive search task, and led to improved performance over the Carnegie Mellon TRECVID 2002 version. Both CMU versions scored extremely well, and with both based on the multi-document storyboard we believe that the storyboard design is critical for the interactive search performance. We believe the browsing interfaces and image-based search improvements made for 2003 led to the increase in performance for the new system, as these strategies allowed relevant content to be found having no associated narrative or text metadata.

Bibliography

[1] B. Adams, A. Amir, C. Dorai, S. Ghosal, G. Iyengar, A. Jaimes, C. Lang, C.-Y. Lin, A. Natsev, M.R. Naphade, C. Neti, H.J. Nock, H.H. Permuter, R. Singh, J.R. Smith, S. Srinivasan, B.L. Tseng, T.V. Ashwin, and D. Zhang. IBM research TREC-2002 video retrieval system. Proceedings of the TREC-11, Gaithersburg, MD, 2002.

[2] B. Adams, C. Dorai, and S. Venkatesh. Toward automatic extraction of expressive elements from motion pictures: Tempo. IEEE Transactions on Multimedia, 4(4):472-481, 2002.

[3] S. Becker, "Mutual information maximization*: models of cortical selforganization*, " Network: Computation in Neural Systems, vol. 7, pp. 7-- 31, 1996

[4] Blum, A., & Mitchell, T. Combining labeled and unlabeled data with co-training. COLT'98

[5] D. Bordwell and K. Thompson. Film Art: An Introduction. McGraw-Hill, New York, USA, 1997.

[6] L. Breiman. Bagging predictors. Machine Learning, 24(2):123-140, 1996.

[7] R. Brunelli and D. Falavigna. Person identi_cation using multiple cues. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(10):955-966, 1995.

[8] J.P. Callan, W.B. Croft, and S.M. Harding. The inquery retrievl system. In *Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain, 1992. Springer.

[9] C.-C. Chang and C.-J. Lin. LIBSVM:a library for support vector machines, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[10] Christel, M.G., and Huang, C. Enhanced Access to Digital Video through Visually Rich Interfaces Proceedings of the IEEE International Conference on Multimedia and Expo (ICME) Baltimore, MD, July 2003), pp. III-21 - III-24.
[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
[12] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI broadcast news transcription system. Speech Communication, 37(1-2):89-108, 2002.
[13] G. Iyengar, H.J. Nock, and C. Neti. Audio-visual synchrony for detection of monologues in video. In IEEE International Conference on Multimedia & Expo, pages 329-332, Baltimore, USA, 2003.
[14] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4-37, 2000.
[15] R.S. Jasinschi, N. Dimitrova, T. McGee, L. Agnihotri, J. Zimmerman, D. Li, and J. Louie. A probabilistic layered framework for integrating multimedia content and context information. In IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2057-2060, Orlando, FL, 2002.
[16] Jin, R. ,Qi, Y., and Hauptmann, A., A probabilistic model for camera zoom motion detection, The sixteenth conference of the International Association for Pattern Recognition (ICPR 2002) Québec City,Canada August 11-15 2002
[17] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML)*. Springer, 1998.
[18] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classi_ers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3):226-239, 1998.
[19] S. Kumar and M. Hebert, "Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field," in proc. IEEE CVPR, Vol. 1, pp. 119-126, 2003.
[20] W.-H. Lin and A. Hauptmann. News video classification using svm-based multimodal classifiers and combination strategies. In *Proceedings of the Tenth ACM International Conference on Multimedia*, Juan-les-Pins, France, December 1-6 2002.
[21] U. Manber and S. Wu, ``Approximate String Matching With Arbitrary Costs for Text and Hypertext," *Proc. of the IAPR International Workshop on Structural and Syntactic Pattern Recognition*, Bern, Switzerland (August 1992), pp. 22-33.
[22] M.R. Naphade, I.V. Kozintsev, and T.S. Huang. A factor graph framework for semantic video indexing. IEEE Transactions on Circuits and Systems for Video Technology, 12(1):40-52, 2002.
[23] Nigam, K. and Ghani, R. Understanding the Behavior of Co-training. In Proceedings of KDD-2000
[24] J.C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Sch¨olkopf, and D. Schuurmans, eds, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.
[25] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
[26] G.M. Quenot, D. Moraru, L. Besacier, and P. Mulhem. CLIPS at TREC-11: Experiments in video retrieval. In Proceedings of the 11th Text Retrieval Conference, Gaithersburg, USA, 2002.
[27] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513– 523, 1988.
[28] T. Sato, T. Kanade, E.K Hughes, M.A Smith, and S. Satoh. Video OCR: Indexing digital news libraries by recognition of superimposed caption. ACM Multimedia Systems, 7(5):385-395, 1999.
[29] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. IEEE Multimedia, 6(1):22-35, 1999.
[30] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. International Journal of Computer Vision. To appear.
[31] C.G.M. Snoek and A.G. Hauptmann, Learning to Identify TV News Monologues by Style and Context, Carnegie Mellon University Technical Report, CMU-CS-03-193, October, 2003.
[32] C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. Multimedia Tools and Applications. To appear. http: //www.science.uva.nl/~cgmsnoek/pub/mmta.pdf.
[33] C.G.M. Snoek and M. Worring. Multimedia event based video indexing using time intervals. Tech. Report 2003-01, Intelligent Sensory Information Systems Group, University of Amsterdam, 2003.
[34] TRECVID 2003 Guidelines. http://www-nlpir.nist.gov/projects/tv2003/tv2003.
[35] V.N. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag, New York, USA, 2000.

[36] J. Vendrig and M. Worring. Interactive adaptive movie annotation. IEEE Multimedia, 10(3):30-37, 2003.
[37] D.H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.
[38] H.D. Wactlar, M.G. Christel, Y. Gong, and A.G. Hauptmann. Lessons learned from building a terabyte digital video library. IEEE Computer, 32(2):66-73, Feb. 1999.
[39] R.Yan, A. Hauptmann and R. Jin, **Multimedia Search with Pseudo-Relevance Feedback,** in *Video Minin,* Rosenfeld, A., Doermann, D., and DeMenthon, D. (eds), Kluwer, Boston, pp. 309 - 338, 2003.
[40] M.Turk and A.Pentland. Eigen Faces for Recognition. Journal of Cognitive Neuroscience, 3(1), 1991.
[41] J. Carbonell, G. Geng, and J. Goldstein. Automated query-Relevant Summarization and Diversity-Based Reranking. In IJCAI-97 Workshop on AI and Digital Libraries, 1997.
[42] R.Yan and A. Hauptmann, **The Combination Limit of Video Retrieval**, ACM Multimedia (MM2003), Berkeley, CA, USA, 2003.

Appendix: Expanded keywords for the 25 manual retrieval topics

ID	Keywords used for manual search
100	City
101	NBA basketball net "Michael Jordan" Lakers
102	pitcher baseball batter yankee MLB
103	"Yasser Arafat"
104	airplane aircraft plane airline airport airways continental
105	helicopter
106	Tomb Cemetery "Arlington National Cemetery"
107	rocket missile warhead
108	Mercedes Benz Logo
109	tank kuwait troops
110	diver diving
111	locomotive train railroad railway Amtrack metro
112	flame fire burn
113	"snow mountain" "mountain climb" "mountain Everest"
114	Osama Bin Laden
115	road traffic block
116	Sphinx
117	crowd riot strike panic
118	Mark Souder
119	Morgan Freeman
120	dow jones gain
121	coffee mocha cappuccino espresso starbucks coffee-mate
122	cats pets
123	" John Paul" Pope
124	"White House"