Confounded Expectations: Informedia at TRECVID 2004

A. Hauptmann, M.-Y. Chen, M. Christel, C. Huang, W.-H. Lin, T. Ng,
N. Papernick, A. Velivelli, J. Yang, R. Yan, H. Yang, and H.D. Wactlar

Abstract

For TRECVID 2004, CMU participated in the semantic feature extraction task, and the manual, interactive and automatic search tasks. For the semantic features classifiers, we tried unimodal, multi-modal and multi-concept classifiers. In interactive search, we compared a visual-only vs a complete video retrieval system using visual AND text data, and also contrasted expert vs novice users; The manual runs were similar to 2003, but they did not work comparably to last year, Additionally, we shared our low-level features with the TRECVID community.

Overview

We first describe the low-level features, which formed the input for all our analysis and which were distributed to other participants. Then we sketch out the experiments done for the semantic features, followed by the search task experiments (manual, automatic and interactive).

Low-level 'raw' features

Low-level features are extracted for each shot. "Low-level" means the features which are directly extracted from the source, videos. We use the term 'low-level' to distinguish them from the TRECVID high-level semantic feature extraction task. The low-level features are derived from several different sources: visual, audio, text and 'semantic' detectors, such as face detection and Video OCR detection. In TRECVID 2004, we extract 16 low-level raw features for the whole data set. This data was provided to all participating groups to encourage other researchers to use or compare their approaches with a standardized feature set.

Image features

A shot is the basic unit in our system; therefore, we extract one key-frame within each shot as a representative image. Image features are then based on the features extracted from that representative image. There are 3 different types of image features: color histograms, textures and edges. For all image features, we split the image into a 5 by 5 grid that tries to capture some spatial locality of information. The distributed data lists the features for each grid cell by rows, starting at the top.

HSV, RGB, HVC and HCSqr Color Histograms

Three different color spaces are used to construct color histogram features: HSV, HVC and RGB. Each grid presents its color histogram in 125 dimensions. Each channel is represented by 5 dimensions and plotted in a 3D histogram. Therefore, for each image, the dimension of the color histogram is 3125 (5*5*125). Due to this high dimensionality, we also provide the mean and variance for each grid and reduce the dimension to 50 (5*5*2). We also add an alternative feature called hcsqr, which is derived from the HVC color histogram, but removes variance and linearizes Hue and Chroma into a 2D histogram.

Texture

Images are first gray-scaled. Each image is convolved with six orientated Gabor filters. For each filter, the image is divided by 5 by 5 grids. The resulting filtered grids are then threshold and reduced to 16 bins histogram. The dimension is 2400 (6*16*5*5).

Edge

The edge detection is done using Canny edge detection. The result of Canny edge detection is convolved with 8 orientations. For each grid, there are 8 dimensions which show the mean magnitude for the 8 orientations. The dimensionality is thus 200 (5*5*8).

Audio features

We extract audio signal every 20 msecs (512 windows at 44100 HZ sampling rate). However, the basic unit of analysis is a shot which has variable length. We therefore calculate the mean and variance for each shot.

FFT

FFT is based on the Short Time Fourier Transform (STFT). The features are the means and variances of the spectral centroid, rolloff, flux and zerocrossings. Another feature called low energy is added. Therefore, they are 9 (4*2+1) dimensions.

MFCC

MFCC features are based on 10 Mel-Frequency cepstral coefficients.

SFFT

SFFT is a simplified FFT. It only lists the mean for the spectral centroid, rolloff, flux and zerocrossings.

Motion features

Motion features try to capture the movement within the shot. Although they very noisy, motion features potentially allow us to move from still image analysis to analysis of the moving video. Since the video was encoded with different MPEG encoders using different motion block numbers, we did not use the MPEG P-frame motion blocks.

Kinetic Energy

Kinetic energy measures the pixel variation within the shot. We convert the image to gray level and calculate the frame by frame differences for every shot. The value is the mean of the difference within the shot. Although we could still split it into 5 by 5 grids, we utilized this feature as a measurement of the stability of a shot and did not split the image.

Optical Flow

The optical flow motion is calculated for every 5 frames. It’s 5 by 5 and each grid contains 3 dimensions. The 3 dimensions are the mean x direction, mean y direction and variance of the magnitude.

Text features

The text feature is derived from the audio transcript.

Semantic Detector features

Two of our features are not quite “low”-level features. However, they are very basic measurements that discover peculiar characteristics in the video. Since people play important roles in news video, face detection gives us useful information. VOCR (video optical character reader) often shows people names and locations.

Faces

The face result collects the information of the most confident face detection result within the shot. The 5 dimensions are confidence, face size, face pose (1 is front, 2 is left, 3 is right), x coordinate of center point, and y coordinate of center point. Center point is the center location of the face detection box.

VOCR

VOCR result generates the information about VOCR detection boxes. The VOCR detection box is the detection result which shows the possible places to contain VOCR. The four dimensions are the number of boxes, average size of boxes, mean of x and mean of y coordinate. The time-stamped contents of the recognized text are listed in a separate file.

High-level semantic features

To classify the 10 high-level semantic features (Boat/Ship, Madeleine Albright, Bill Clinton, Train, Beach, Basketball scored, Airplane takeoff, People walking/running, physical violence and Road), our baseline was a single modality classification approach. For each, we chose one single feature set of our standard low-level feature classes and built a classifier for that feature. Separate runs used a multi-modality classification strategy. Here, we built classifiers for each low-level feature and then combined them with a meta-classifier (stacking).

The main challenge for semantic feature extraction from video is the large diversity. Low level features tend not to capture a complete semantic class well. For example, ‘outdoors’ contains many different concepts, as well as colors, textures and shapes. It may be an urban scene, a rural scene, a beach scene or another natural scene, each with different colors, textures, etc. Therefore, we attempted to utilize other semantic features to boost the performance of a specific classifier (or detector). For example, if we want to build a classifier for outdoor scenes, we can build (hopefully easier) classifiers, like sky, ocean, tree, grassland, road, building and other outdoor-related concepts. Ideally, the outdoor classifier can gain power from other easy and strong detectors, like sky, ocean and grassland, and thus the outdoor detector will be able to correctly classify many different scenes.

Our approach in TRECVID 2004 was to find other concepts related to the classification task. Then, using the principles of causation and inference, we developed two classification strategies.

ID	**Target Concept**	**Causally related concepts**
28	Boat/Ship	Boat, Water_Body, Sky, Cloud
31	Train	Car_Crash, Man_Made_scene, Smoke, Road
32	Beach	Sky, Water_Body, Nature_Non-Vegetation, Cloud
33	Basket Scored	Crowd, People, Running, Non-Studio_Setting
34	Airplane Takeoff	Airplane, Sky, Smoke, Space_Vehicle_Launch
35	People Walking/running	Walking, Running, People, Person

36	Physical violence	Gun_Shot, Building, Gun, Explosion
37	Road	Car, Road_Traffic, Truck, Vehicle_Noise

Table 1. Causal relationships for the TRECVID 2004 concepts.

Causation

The common annotation set, distributed in TRECVID 2003, labeled several hundred semantic concepts in TRECVID 2003 development set of 47322 shots. Among those concepts, there are 190 concepts which have a frequency higher than 10. We analyzed the causal relationship of these 190 concepts to 8 high-level semantic features concepts, listed above We excluded Madeleine Albright and Bill Clinton, because the latter were more suitable for specific person x search strategies describe below. We selected the top 4 causal origins for each concept and grouped them together. Table 1 shows the respective 4 concepts which were determined to cause the evaluated target concepts.

Inference

Using each group of 5 concepts (4 causal ones and the target concept), we built a multi-modality classifier for each concept. To train the combination parameters, we split our training data into two sets. The first set is used to build the classifiers for each individual concept. The second set is used to validate the combination. The next step is to infer the causational concept classifier results into target concept. We then experimented with two approaches (A and B) to combine the classifier results.

A- We use the confidence of causal relationship (form 0 to 1) and the error rate obtained on the training set to combine the results.

$$s' = s + \sum_{S_i \in causation(s)} cw_i * (1 - cerror_i)^6 * s_i \qquad (1)$$

where s is the multi-modality result for the target classifier, cw is the causal relationship from the causal model, cerror is the classifier error for this concept classifier in a validation set, s_i is the multi-modality result for this casual concept. The power 6 was obtained from the validation set.

B – For the second approach, we again build individual concept classifiers from the first training set. Then we apply logistic regression on the result over the validation set. The final score for the target concept will by the combination of the 5 concept classifiers using logistic regression. Table 2 shows results for both methods and the baseline (multi-modality classifier result).

	28	31	32	33	34	35	36	37	
A linear	0.081	*0.001*	0.031	*0.517*	0.030	0.008	0.002	0.046	0.090
B	*0.110*	0.000	*0.039*	0.503	*0.035*	*0.099*	*0.003*	*0.062*	*0.106*
Baseline	0.137	*0.001*	0.023	*0.517*	0.014	0.008	0.002	0.045	0.093

Table 2. Comparison between different approaches. *Red* indicates the best result of the 3.

Person Finding in Broadcast News Video

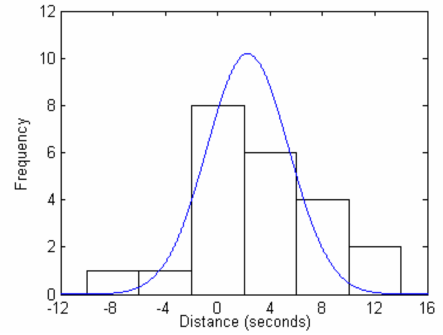

Figure 1: Distribution of shots containing "Madeline Albright" w.r.t. the occurrence of her name

There are two feature extractions tasks in TREC04 with people, namely finding Madeline Albright and President Clinton. Due to the distinctness of these two tasks, we believe that *specialized* methods are needed for person finding. Obviously, the most important evidence for that is the occurrence of the person's name in the video transcript, which makes text IR approach a natural choice for person finding. However, this method suffers a serious drawback caused by the *temporal mismatch* between names in transcript and the people in video frames. For example, in a news story about Clinton, his name is first mentioned by the anchor without his face being shown, and after the anchor shot Clinton appears but his name does not appear again. Figure 1 shows the frequency of the shots containing Madeleine Albright at certain distances from the point where her name appears in the transcript. We model this distribution using a Gaussian (i.e, $p(D(shot\ ,name\ )) = N(u,\sigma^2)$, where D is the distance), whose parameters are trained using Maximum Likelihood from the development set. To overcome the temporal mismatch problem, we can propagate the IR score of the shot having the name (say, S_0) to its neighboring shots (say, S_i) according to this distribution. Therefore, the updated IR score of a shot S_i is $R(S_i) = R(S_0)\int_{end\ (S_i)}^{start\ (S_i)} N(u,\sigma^2)$, where $start(S_i)$ and $end(S_i)$ are the distance from the start or end of S_i to the position of the name, respectively.

Another problem particularly notable for Bill Clinton is that, sometimes a person's name is mentioned in a story but the person never appears in the proximity. As a typical example, a news story mentions the policy of "Clinton administration", where Clinton himself never appears. To address this problem, we examine all the bigrams around "Clinton" and for each bigram compute the conditional probability *P(Clinton shows up | bigram$_i$)* that Clinton appears in the proximity of a bigram. This probability is integrated into the IR score by $R'(S_i) = R(S_i)\frac{1}{N_i}\sum_{j \in N_i} P_{bigram\ j}$, where N_i are the bigrams around shot S_i.

Modalities other than the transcript also provide useful information for person finding. For example, facial similarity between the face detected from the shot and a pre-built eigenface model of the target person is another important clue for determining whether the shot has the person. Moreover, since anchor shots rarely contain the target person, excluding them from the results help reduce many false alarms, as long as we have an accurate anchor detector. The results of commercial detector, reporter detector, and weather forecast detector are useful in a similar way. Since each evidence results in a score for a shot (including text retrieval), we combine these scores in a linear way to get the final score, with the weights for each evidence trained by logistic regression based on the truth manually collected on development set for each person (Albright and Clinton).

Automatic and Manual Search from Multimodal Features

Besides a few submissions using pure textual information, most of our automatic and manual search methods are the variations of the same retrieval framework, which re-ranks the results of text (transcript)-based retrieval based on multimodal evidences using a linear model. These methods differ on how the re-ranking weights are trained. The performance of these methods are compared and analyzed.

Multimodal Evidences for Video Retrieval

The aim of video retrieval is to find a set of video shots for a given query, which is formulated in multi-modalities including free text description, example images, and example video snippets. There exist multiple sources of evidence that suggest the relevance between a shot and a query, including:

- ***Text retrieval*:** The match between the textual query and the portion of video transcript (synchronized closed-captions plus speech recognition) corresponding to the shot provides the key evidence on the relevance of the shot. An IR search engine based on TF*IDF weighting scheme is adopted to generate the text retrieval scores for shots. However, as discussed in Section 0, a relevant shot does not always have keyword hit on itself; more often the keyword hit is on its neighboring shots. Similarly, we overcome this temporal mismatch by propagating IR score of a "hit" shot S_0 to its neighboring shots S_i in a window by an exponential decay function, i.e., $r(S_i) = r(S_0) \cdot \alpha^i$, where α is within [0,1]. Thus, the closer the shot is to the position of keyword hit, the larger score it gets.

- ***Content-based image similarity***: The similarity of the key-frame of the shot to the example images is another clue to the relevance of the shot. In our system, three image similarity scores are computed using 150-d HSV color histogram, 108-d Gabor texture filters, and 200-d Canny edge feature, respectively. For data consistency reason, the example images used are the key-frames of the provided video examples, while the provided (external) image examples are discarded as they are not from news videos.

- ***Specific shot detection:*** There are certain types of shots that rarely appear in the results, among which are anchor shots, weather-forecast shots, and commercial shots. We built specialized detectors to estimate the probabilities of a given shot being one of the three shot types. These probability scores are useful in eliminating false alarms.

- ***Face similarity*:** For queries for finding a specific person, the similarity of detected face(s) from the shot with a pre-trained face model of the person is very important.

A Re-ranking Retrieval Framework

For each candidate shot s_i for a given query, we can obtain a set of similarity scores computed from different sources of evidences listed above, denoted as $\{r_{ij}\}_{j=1,\ldots,M}$, which have been normalized into rank-base probability scores (i.e., within [0,1]). It naturally follows the problem of combining these scores in an optimal way. We adopt a linear combination of these scores as $R_i = \sum_{j=1}^{N} w_j r_{ij}$, where R_i is the final score for shot s_i, and w_j is the weight for *j*th component score. However, the relevant shots of a typical shot constitute only a tiny portion of all the shots, which makes the weight-learning a difficult and expensive *rare-class learning* problem. To mitigate this problem, we reduce the training dataset from the whole set of shots into the set of shots returned by text retrieval, since text retrieval is able to obtain most of the relevant shots (i.e., high recall). Accordingly, the retrieval is conducted in the following steps:

(1) Compute the similarity scores (including the text retrieval scores) from multi-modal evidences for all the shots.

(2) Obtain a set of top-*N* shots ranked according to text retrieval score; (in our system, *N* = 400)

(3) Re-rank the *N* shots by integrating the component similarity scores computed from multimodal evidences using the weights trained, and put re-ranked shots into the top of the final result list;

(4) Rank the remaining shots based on multimodal evidences and put the top (1000-N) shots into the bottom of the final result list (so there is 1000 shots in the list).

We set *N* less than 1000 (e.g., the limit of shots for each query) so that the shots that are not matched by text retrieval have a chance to appear in the result. This is particularly useful if the image examples are accurate representations of the relevant shots. We name this as a "*re-ranking*" framework since it firsts find the proximities of the results by text retrieval and then re-ranks the shots returned by text retrieval with the multimodal information. Two issues need to be addressed as to the re-ranking weights used.

- *Weight model*: The result of text retrieval features high recall but low precision, i.e., it contains most of the relevant shots but also many irrelevant shots. We have two opposite assumptions on the text retrieval scores: (a) Among the shots returned by text retrieval, in average, the relevant shots have higher scores than the irrelevant ones; and (b) The relevant ones do not necessarily have higher scores. Given assumption (a), we not only use text retrieval to find the top-N candidate shots but also integrate the text retrieval scores as an evidence in the re-ranking process (i.e., the weight for text retrieval is non-zero). This weight model is named as "*text sensitive*". In contrast, given assumption (b), we only use text retrieval to find the top-*N* candidate shots, but exclude its scores from the re-ranking process (i.e., its weight is set to zero). Accordingly, this weight model is described as "*text insensitive".*

- Weight learning strategy: The weights can be trained based on queries with ground-truth using different strategies, including:

1) learning weights for each pre-defined query type based on handcrafted sample queries (and truth) and the development set (Section 0);
2) learning weights specific to each TREC04 query based on manually collected truth on the development set (Section 0); and
3) learning weights specific to each TREC04 query based on pseudo-feedback examples on the testing set (Section 0).

Learning Query-Type-Specific Weights and Query Type Classification

Video queries can be classified into a few semantic types, and we assume that queries of the same type are likely to have similar "ideal" combination weights. Therefore, it is reasonable to train a set of weights for each query type, rather than for each single query (which can be expensive and unreliable given the limited training data). We define the following 5 query types:

- **Named Person:** finding a named person, possibly with actions, e.g., "*Boris Yeltsin".*

- **Specific Object**: queries for a specific object with a unique name, which distinguishes this object from others of the same type, e.g., "*Zooming in on the US Capitol Dome".*
- **General Object**: queries for a certain type of objects, which can be modified by adjectives, number, etc, e.g. "*Find shots of one or more bicycles rolling along".*
- **Sports**: queries for a scene related to a sports event, such as "*Find shots of a tennis player contacting the ball with his or her tennis racket".*
- **Scene**: queries depicting multiple types of objects in certain spatial relationships, e.g., "*Fingers striking the keys on a keyboard which is at least partially visible".*

To train the weights for each query type defined in previous subsection, we created a set of 40 artificial queries with 6-10 queries in each type, as shown in Appendix II, and manually collected their truth in the development set. For each query q^k we obtain a set of candidate shots $\{s_i^k\}_{i=1,...,N^k}$ by text search, and for each shot s_i^k we compute a set of similarity scores as $\{r_{ij}^k\}_{j=1,...,M}$. We treat the score vector for each shot as a data item, and its label is set to 1 if the shot is a relevant shot for the query and 0 otherwise. We pool the score vectors of the queries of query type T and train a set of combination weights $\{w_j^T\}_{j=1,...,M}$ using logistic regression.

To apply the query-type-specific queries to new queries, it is desirable to classify a query automatically into one of the five query types. Our query classification process consists three phases: *query focus analysis, natural language (NL) analysis* and *rule-based classification*. Firstly, we determine query focus, the query concept that embodies the information expectations expressed by the query [Lehnert 1978]. Instead of applying sophisticated parsing and disambiguation techniques, we identify query focus as *the first noun or base noun phrase in the sentence after removing question header*. Question headers are removed based on surface text patterns. For instance, we have the question headers and query foci for the following queries: *[Find the clips talking about]$_{question\_header}$ [water project]$_{query\_focus}$*.

Secondly, we perform NL analysis on the queries after removing their question headers. It consists part-of-speech (POS) tagging, shallow parsing (which identifies base noun phrases), and named entity (NE) recognition (which identifies person, organization and location). Lastly, based on query focus, NL analysis results and several special word lists generated from WordNet [Fellbaum 1998] such as words about sports, people and video terminology, we apply rules to classify the queries. If the query focus contains person entities, we classify the query into class "person". For the rest of queries, whose query foci containing sport word, we classify them into "sports". If their query foci contain location or organization entities, we classify them into "specific object". In another case, if their query foci contain people words and there are sport words in the rest of the sentence, the query goes to class "sports" as well. Similar case happens in class "specific object". For instance, *"Clips of people who are water skiing"* goes to "sports" while *"Find a person (an expert) showing the aircraft X-29"* goes to "specific object". For the rest of queries, we count the number of nouns and base noun phrases after removing question headers and stop-words. If this number is 1, we classify the query as "general object", if greater than 1, "others". For example, *"Find shots of a mug or cup of coffee"* goes to "general object", while *"Find shots of one or more roads with lots of vehicles"* goes to "others".

Based on our experiments, this query classification method achieves 100% accuracy on the TREC 2004 queries. Once a query's type is predicted, we can use the corresponding set of weights to generate the results for that query.

Learning Query-Specific Weights based on Manual Annotation

Queries of the same type can be still different from each other. To model the idiosyncrasies of each query it is desirable to train the combination weights for each query. In manual search, since for a query q^k a user has 15 minute to reformulate the query, we let the user use this time to collect relevant shots (as many as possible) of q^k in the development set. The collected truth of each query, though incomplete, is used to train the combination weights $\{w_j^k\}_{j=1,\dots,M}$ *specific* to q^k by logistic regression. The weights are then used to re-rank the candidate shots of q^k in the testing set to generate the final rank of shots. We expect this approach to achieve higher performance than using the query-type-specific weights, if (1) the distribution on the development set and testing set is similar, and (2) sufficient training data (i.e., relevant shots) are collected within 15 minutes.

Learning Query-Specific Weights based on Co-Retrieval

Similar to last year, we use the idea of "Co-retrieval" to train query-specific weights from pseudo relevance feedback, which is done automatically without human effort. Instead of having users annotate the truth, pseudo feedback tries to "guess" the relevant and irrelevant shots of a given query based on certain clues. Since text retrieval provides the most important clues, our strategy is to among the top-400 shots ranked by text retrieval scores, label the first 100 shots as relevant and the rest 300 as irrelevant. The combination weights are trained based on these 400 pseudo-labeled examples by logistic regression, and then used to re-rank the candidate shots of that query. Note that since the pseudo-labels are from text retrieval scores, if we include text retrieval when training the "text-sensitive" weights, its weight will be infinite. So in this case we manually set the weight for text retrieval to a fixed value, while the weights for other features are still trained in the same way.

Submissions and Experiments

We submitted 8 manual runs and 2 automatic runs, as summarized in

Submission ID	**Submission Type**	**Textual Query**	**Feature Used**	**Weight Learning Strategy**	**Training Queries**	**Training Data**	**Weight Model**
M_C_2_05M_5	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online manual labeling (15min) on development set	Text sensitive
M_C_2_06M_6	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online manual labeling (15min) on development set	Text insensitive
M_C_2_07M_7	Manual	Manually expanded	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text sensitive
M_C_2_08M_8	Manual	Manually expanded	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text insensitive
M_C_2_09M_9	Manual	Manually expanded	Text	N/A	N/A	N/A	N/A
M_C_2_10M_10	Manual	Manually expanded	ASR	N/A	N/A	N/A	N/A
M_C_2_S3A_S	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online pseudo-feedback on testing set	Text insensitive
F_C_2_S4A_S	Automatic	Original	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text insensitive
F_C_2_S5A_S	Automatic	Original	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text sensitive
M_C_2_S6A_S	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online pseudo-feedback on testing set	Text sensitive

Table 1. For manual runs, we expand each query with keywords that are manually chosen within 15 minutes by examining the development set (to see which keyword can find the relevant shots) using Informedia Client. The reformulated queries are listed in Appendix I. For automatic runs, the original queries after striping off the common head (e.g., "Find shots of") are used. Also note that by the "text" we used is the combination of synchronized closed caption plus the LIMSI automatic speech recognition results.

Submission ID	**Submission Type**	**Textual Query**	**Feature Used**	**Weight Learning Strategy**	**Training Queries**	**Training Data**	**Weight Model**
M_C_2_05M_5	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online manual labeling (15min) on development set	Text sensitive
M_C_2_06M_6	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online manual labeling (15min) on development set	Text insensitive
M_C_2_07M_7	Manual	Manually expanded	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text sensitive
M_C_2_08M_8	Manual	Manually expanded	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text insensitive
M_C_2_09M_9	Manual	Manually expanded	Text	N/A	N/A	N/A	N/A
M_C_2_10M_10	Manual	Manually expanded	ASR	N/A	N/A	N/A	N/A
M_C_2_S3A_S	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online pseudo-feedback on testing set	Text insensitive
F_C_2_S4A_S	Automatic	Original	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text insensitive
F_C_2_S5A_S	Automatic	Original	Text & Multimodal	Query-type-specific	40 sample queries	Offline manual labeling on development set	Text sensitive
M_C_2_S6A_S	Manual	Manually expanded	Text & Multimodal	Query-specific	TREC04 queries	Online pseudo-feedback on testing set	Text sensitive

Table 1: Descriptions of manual and automatic search submissions

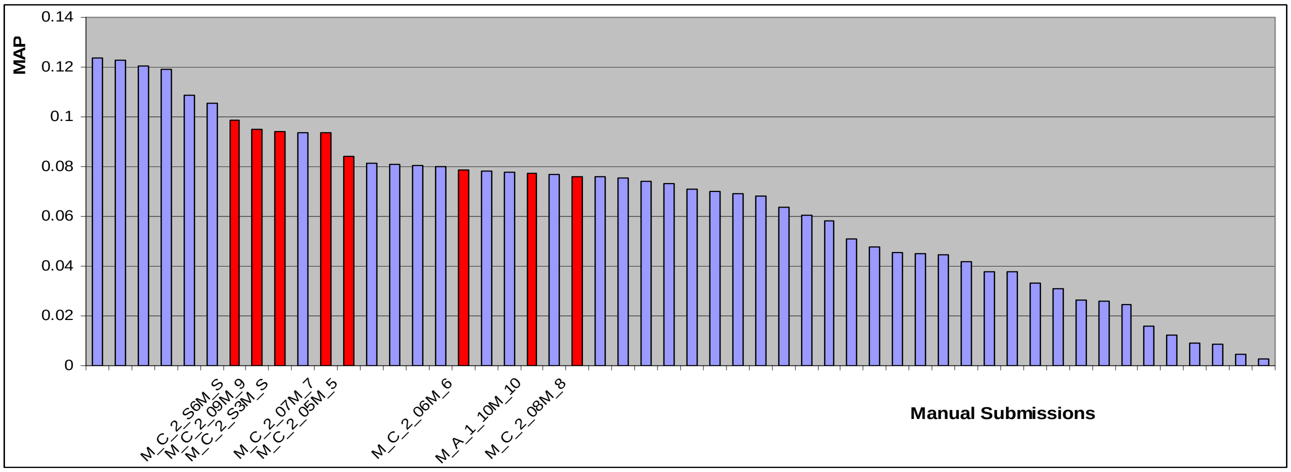

Figure 2: Performance of CMU's manual search submissions versus other manual submissions

Experimental results of the MAPs (mean average precision) of our submissions against other submissions are shown in **Error! Reference source not found.** for manual search and in Figure 3 for automatic search. There are some interesting observations from the manual runs:

- For manual search, the text-only run M_C_2_09M_9 achieves higher MAP than most of the runs that use multimodal information. This shows that text retrieval provides the dominant evidence, while other evidences are relatively weak and noisy.

- Surprisingly, our best manual search performer M_C_2_S6A_S uses query-specific weight trained on the (supposedly unreliable) pseudo-feedback. It outperforms the methods using query-specific weights trained on manually collected truth in development set. This implies that the data distribution between development and testing data is different, which makes the weights trained on development set misleading.

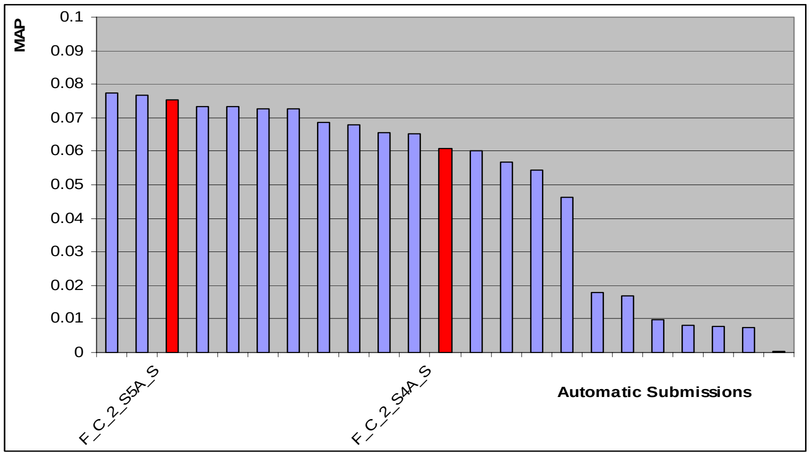

Figure 3: Performance of CMU's automatic search submissions versus other automatic submissions

- Between each pair of "text sensitive" and "text insensitive" methods using the same learning strategy, the "text sensitive" one always beats the other by a significant gap. This shows that the score of text retrieval is a good indication of the degree of relevance of a shot.

- The MAP of our best automatic run is about 25% less than the MAP of our best manual run. We believe that the difference is mainly due to the use of original queries instead of manually formulated queries.

- For automatic search, we find the idea of using query-type-specific weights effective, since our best run differs from the overall best runs only by a small gap.

Interactive search:

We wanted to compare a single expert's performance with full system to prior years' tests (TRECVID 2003, TRECVID 2002) so once again we had one run of an expert (single user) with the full system. The system began with the baseline as used in TRECVID 2003 with some added improvements suggested by user studies.

The main goal of TRECVID 2004 CMU interactive search work, though, was to test the effectiveness and usability (including satisfaction) of a visual-only system making no use whatsoever of closed captioning or narrative text from automatic speech recognition. A within-subjects study was conducted with 24 CMU students to determine differences in video information retrieval effectiveness and user satisfaction between a full-featured system and a "visual-only" system. To complete the comparisons, a set of experts, each doing 4 topics like the CMU novice users, were also recruited to use the visual-only system, enabling comparisons of novice vs. expert interactions with the visual-only system. The visual-only system contained no information from closed captioning or automatic speech recognized text.

So, the CMU interactive search made use of 4 trials, essentially: expert with full system, experts with visual-only system, novice with full system, novices with visual-only system. The within-subjects portion of the experiment was actually run twice through the TRECVID topics (24 users rather than 12, each doing 4 topics 15 minutes each topic), so in actuality we submitted 6 runs: the four enumerated above, plus a supplemental novice full and supplemental novice visual-only.

System had the following features: text search, image-based search (color or texture), "best-of" prebuilt concept set, e.g., best roads. In prior TRECVID years there was some thought given to making the feature detection task (concept detection) be a building block for search tasks. However, this year the features were overall very focused, essentially becoming search tasks in themselves. Features like basketball going through the hoop or Madeleine Albright were unlikely to be of use to general querying, or practically were unlikely to be available in a generic video retrieval system. Features like indoor, outdoor, face, city, etc., are more general and do have utility in interfaces such as the feature filtering interface (see PPT set for example). For this year, only beach and road were used and available as a "best-of" set for the user in interactive search.

Slide 12: hoped to find some topics for which visual-only search produced better results than full systems, e.g., perhaps visual-only strategy would be overlooked when great text search is available but the visual-only strategy would have been highly productive.

Slide 13-14: This did not happen: full system outperformed the visual-only system across almost all topics for both experts and novices

Slide 15: Interactive search better than manual search which is better than automatic search. For our study, expert with full system significantly better than novice with full system. Full system significantly better than visual-only system. Experts with visual-only system performed better than novices with visual-only system (but difference is tighter than with full system).

A note of caution: By increasing the number of users per topic, and using a simple decision strategy such as pick the user for a topic who submits the most shots, the performance for that topic can be improved over just having a single user per topic. There of course will be a ceiling effect, but for the first time Carnegie Mellon had an interactive search where two (or more, but in this case two) users answered the same topic with the same interface treatment. The mean average precision for the two runs through the 23 TRECVID topics (throwing out topic 146) are nearly identical: 0.245 mean average precision (MAP) for first run with the "Full System", 0.249 MAP for second run with the "Full System". 0.099 MAP for first run with the "Visual-only System", 0.103 MAP for second run for "Visual-only".

However, by using the decision strategy noted above the topic answers can be partitioned into a submitted set with a higher MAP and the remaining answers put in a supplemental set with a lower MAP. The interaction logs showed that text search was used differently: expert with full system more precise, spends time analyzing returned results. Experts in the visual system have less "0 results" text queries; also inspect results and hence issue fewer queries. For visual-only systems, experts are willing to use image search and try the precomputed semantic feature sets, novices are not. The Visual-only users must by necessity use image queries more. Overall, even visual-only makes strong use of text search, as VOCR text was the only target text available for this text search, no CC or ASR was available in this condition.

Analyzing the answers for the interactive conditions, we found that

- Correct answers are strongly tied to the text query. A text query is the ideal strategy for a news corpus. Novices will need encouragement to use advanced browsing and querying features like image search and precomputed sets.
- The relative high information retrieval performance by both experts and novices is due to its reliance on an intelligent user possessing excellent visual perception skills to compensate for comparatively low precision in automatically classifying the visual contents of video
- Visual-only interactive systems better than full-featured manual or automatic systems

- ASR and CC text enable better interactive retrieval
- Novices will need additional interface scaffolding and support to try interfaces beyond traditional text search.

References

[Lehnert 1978] W. G. Lehnert. 1978. “The Process of Question Answering: a computer simulation of cognition”. Lawrence Erlbaum Associates.

[Fellbaum 1998] Christiane Fellbaum. 1998. “WordNet: An Electronic Lexical Database”, http://www.cogsci.princeton.edu/~wn/.

[Gauvain78] J.L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. Speech Communication, 37(1-2):89-108, 2002.

[Lin03] C.-Y. Lin, B. L. Tseng and J. R. Smith, "Video Collaborative Annotation Forum: Establishing Ground-Truth Labels on Large Multimedia Datasets," NIST TREC-2003 Video Retrieval Evaluation Conference, Gaithersburg, MD, November 2003.

Appendix I: Manually Expanded Queries for Manual Search

Query ID	Keywords of manually expanded query
0125	street pedestrians vehicles cars
0126	flood storm
0127	dogs pets
0128	Henry Hyde
0129	capitol congress republican democrat
0130	hockey rink NHL
0131	keyboard computer laptop
0132	stretcher accident hospital injured wounded
0133	Saddam Hussein
0134	Boris Yeltsin
0135	Sam Donaldson
0136	golf hole golfer birdie "pga tour"
0137	Benjamin Netanyahu
0138	steps stairs stairway staircase
0139	handheld weapon riot gun rifle shoot
0140	bicycles bikes biker
0141	umbrella raining rainy
0142	tennis racket Agassi Sampras ATP WTA
0143	wheelchair
0144	Clinton
0145	horse race
0146	skiers slalom skiing pole "Winter Olympics"
0147	fire flames smoke building
0148	slogan banners march protest demonstration

Appendix II: Sample Queries for Training Query-Type Weights

Q-Type	Query
Named Person	Find shots of Madeleine Albright
	Find shots of Kenneth Starr
	Find shots of David Kendall
	Find shots of Bill Gates
	Find shots of Slobodan Milosevic
	Find shots of Monica Lewinsky
	Find shots of Yasser Arafat
Specific Object	Find shots of Tomb of the Unknown Soldier at Arlington National Cemetery
	Find shots of Mercedes logo
	Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day
	Find shots of the front of the White House in the daytime with the fountain running
	Find shots of Volkswagen Beetle car
	Find shots of the logo of American Airline
	Find shots of the logo of Microsoft Windows
	Find shots of the facade of Supreme Court in Washington DC
General Object	Find shots of an airplane taking off
	Find shots of a rocket or missile taking off
	Find shots of a locomotive approaching the viewer
	Find shots of one or more tanks
	Find shots of flames
	Find shots of a mug or cup of coffee
	Find shots of one or more cats
	Find shots of a helicopter in flight or on the ground
	Find shots of one or more hot air balloons
	Find shots of a flying eagle
	Find shots of a space shuttle on the ground or in the flight
Sports	Find shots of a basket being made - the basketball passes down through the hoop and net
	Find shots behind the pitcher in a baseball game as he throws a ball that the batter swings at
	Find shots of football players
	Find shots showing that someone is doing figure skating
	Find shots of two people playing a tennis game
	Find shots of car race
	Find shots of a soccer game in progress
Scene	Find shots of one or more roads with lots of vehicles
	Find shots of one or more groups of people, a crowd, walking in an urban environment
	Find shots of aerial views containing both one or more buildings and one or more roads
	Find shots of one or more snow-covered mountain peaks or ridges
	Find shots of a flying satellite with the space and the earth (partially) visible
	Find shots of people spending leisure time at the beach (people and beach must be visible)
	Find shots of people walking on red carpet (people and the red carpet must be visible)