Fudan University at TRECVID 2007

Xiangyang Xue, Hui Yu, Hong Lu, Yuefei Guo, Yuejie Zhang, Shile Zhang, Bin Li, Bolan Su, Yingbin Zheng, Wenjian Zhou, Lei Cen, Jie Zhang, Yu Jiang, Jiahui Qi, Jiaojiao Lu, Qian Diao, Zhenzhen Shi, Zichen Sun
Department of Computer Science and Engineering, Fudan University, Shanghai, China

In this notebook paper we describe our participation in the NIST TRECVID 2007 evaluation. We took part in two tasks of benchmark this year including high-level feature extraction and interactive search.

For high-level feature extraction, we submitted 4 runs.

FD_SVM: using SVM.

FD_SVM_BN: using SVM and ontology.

FD_MTL: using multi-task learning.

FD_MTL_BN: using multi-task learning and ontology.

Evaluation results illustrate that there are both advantages and disadvantages exist in all methods.

For search, we submitted 5 interactive runs.

Fudan_P: using multi-model and AP-based fusion.

Fudan_R: using multi-model and MGR fusion.

Fudan_C: cross system retrieval.

Fudan_T: textual retrieval

Fudan_I: image retrieval

Evaluation results illustrate that the AP-based fusion method yields higher precision while the MGR fusion method finds more positive shots than other runs. We also experimented with simple cross system interactive retrieval to estimate the impact of manual browsing on the results.

1. Introduction

In the field of content-based video retrieval, many efforts have been made by developing effective techniques for analysis, indexing, and searching of video from the database. In order to compare different algorithms and systems, TRECVID provides a standard dataset and evaluation criterion annually, which has gathered hundreds of organizations all over the world to participate. This year we continued to join this interesting and challenging work with some new methods, and made quite a few progresses over the past few years in each task we took part in, i.e., high-level feature extraction and interactive search.

2. High-level Feature Extraction

Because of many monochrome video clips in the corpus this year, we focus on the texture features

during the low-level feature extraction from the key frames of shots. And 3 different scales are tried while the feature extraction to capture both of the regional and the global characteristics of the key frames. At the learning stage, we've tried a multi-task learning (MTL) method as well as the regular SVM method. Finally, classifiers trained from each low-level feature are fused with the same weight as our experiences show that this unsupervised way performs well compared to other general fusion methods. However, we make use of Bayesian networks, which modifies the final fusion results, to achieve a better performance. We've submitted 4 runs as shown in Figure 1.

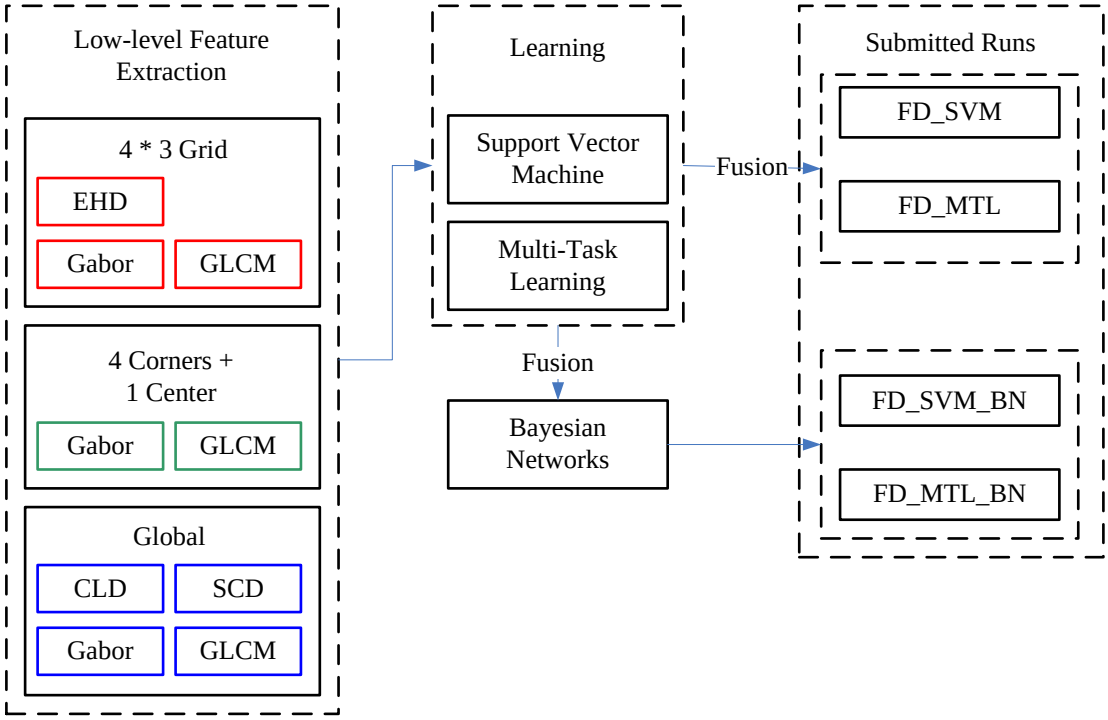

Figure 1 Overview on the framework of high-level feature extraction

2.1. Low-level Feature Extraction

Five kinds of visual features are extracted: Gabor, GLCM (Gray Level Co-occurrence Matrix), EHD (Edge Histogram Descriptor, defined in MPEG-7), CLD (Color Layout Descriptor, defined in MPEG-7) and SCD (Scalable Color Descriptor, defined in MPEG-7). To enhance on characterizing the monochrome shots, texture features, including Gabor and GLCM, are extracted from 2 grid region-based scales besides the global scale. We use GLCM as a filter just like the Gabor filter. The variance and the correlation are calculated as the output of the GLCM filter at each pixel with a 9 * 9 window. On the parameters of the Gabor filter, we take 3 scales and 6 directions. After obtaining the filtered image which is processed by a Gabor or GLCM filter, the first, the second and the third order moments of a specific region are calculated to describe the texture feature of this region.

EHD is based on a grid layout, we extracted the edge histograms at the scale of 4 * 3 grid. Since the major part of the corpus is colorful and the color information is also very important, we use CLD and SCD to describe the global color information, which covers the layout and the distribution of the color in an image.

2.2. Training of SVM

We use the RBF kernel and its optimal parameter pair (C, ɣ) is obtained from a gird search. A binary classifier is trained on a single feature. To deal with the unbalance problem between the number of positive samples and the number of negative samples on many high-level features, we randomly choose a part of negative samples to train the classifiers.

2.3. Multi-Task Learning

The high-level feature retrieval tasks have the following two features: 1) The training datasets for most tasks are very small and the training samples are insufficient to describe the real data distributions, thus we can hardly learn the classifiers which have good generalization ability; 2) The tasks are not mutually exclusive, i.e., a sample can be the positive sample for more than one task, thus different tasks may have potential relatedness which is established by the cooccurrence of different high-level features in the same sample.

We've observed that the tasks can be first clustered based on the human knowledge, and we split 33 tasks into 12 clusters (see table 1). For example, "Airplane" is very likely to appear in the "Sky" setting, so we put them in the same cluster. Given such problem setting, multi-task learning (MTL) may improve the generalization of the classifiers. We learn the classifiers for the tasks in each cluster simultaneously via maximum penalized likelihood; the objective function of the tasks in a cluster is as follows:

$$p(\{\mathbf{y}_l\}|\{\mathbf{X}_l\},\theta)=\left\{\prod_l \int p(\mathbf{y}_l|\alpha_l,\mathbf{X}_l)p(\alpha_l|\theta)d\alpha_l\right\}p(\theta),$$

where $l\in\{1,2,\ldots,L\}$ is the task index in a cluster, $\mathbf{X}_l$ and $\mathbf{y}_l$ are the training data and the corresponding labels for the l th task, α_l is the coefficients for the classification function for the l th task, i.e., $f_l(\mathbf{x})=\sum_n \alpha_l^{(n)}\kappa(\mathbf{x}_n,\mathbf{x})$, and θ is the hyper-prior distribution of f_l. We use the algorithm proposed in [1] to estimate the coefficients $\alpha_l, l\in\{1,2,\ldots,L\}$, and we adopt the RBF kernel.

Table 1 Task Clusters

Cluster	High-level Features
01	Airplane, Sky
02	Boat_Ship, Waterscape_Waterfront
03	Bus, Car, Truck
04	Charts, Maps, Weather
05	Computer_TV-screen, Studio
06	Meeting, Office
07	Court, Flag-US, Building

08	Urban, Road
09	Desert, Military, Natural-Disaster, Explosion_Fire
10	People-Marching, Police_Security, Prisoner
11	Animal, Mountain, Snow, Vegetation
12	Sports, Walking_Running, Crowd

2.4. Bayesian Networks

As we mentioned before, these high-level features are not mutually exclusive. On the other hand, there are close relations between some high-level features, e.g. *Face* and *Person* (a shot which contains at least one face of course contains at least one person), *Urban* and *Building* (an urban scene usually has buildings in it), etc. Such relationships will help much to improve the confidence outputs by the classifiers.

Conditional probabilities are suitable to describe these relationships. We've built a matrix M_{ij}, whose element $m_{ij} = p(C_i \mid C_j)$. C_i denotes the existence of the high-level feature i. This matrix is asymmetric and its diagonal elements are all 1s. For simplicity, we assume the conditions are independent, i.e. $p(C_i \mid C_j, C_k) = p(C_i \mid C_j)\, p(C_i \mid C_k)$. However, the fact is that many high-level features are not independent. Some examples are listed above. Thus we pick a small subset S of high-level features, in which the elements are almost independent. Furthermore, the positive samples of these features are adequate to train good classifiers, which assures the conditions are reliable. The features in the subset S are *Building*, *Face*, *Office*, *Road*, *Sky*, *Vegetation*, *Walking_Running*, and *Waterscape_Waterfront*. For a shot t, the output by the fused classifier on the high-level feature i is $F_i(t)$. The final output for the ranking $\mathbb{F}_i(t)$ is computed as

$$\mathbb{F}_i(t) = F_i(t) + \sum_{j \in S} P\left(C_i \mid C_j\right) F_j(t) \qquad C_i \neq C_j$$

Although we only pick 8 high-level features as conditions, the final result $\mathbb{F}_i(t)$ will be affected too much if all of the 8 high-level features are used. So we only select 2 features that are most relative to C_i, which have the 2 largest $p(C_i \mid C_j)$.

2.5. Evaluation

We've totally submitted 4 runs. From Figure 2, we can see the obvious gap between our best results and the best results from other TRECVID2007 teams. The results show that there remains much work we have to do in the future. Especially such as *Weather*, *Desert*, *Police_Security*, *Military*, *Flag-US*, and *Explosion_Fire*, the best results are more than 10 times better than ours.

In our 4 runs, the SVM runs' results are almost better than the MTL runs'. It may be caused by only using a subset of positive samples for some high-level features that have many positive samples since the MTL method is too exhausting to afford too many training samples. But for some high-level features that have a few positive samples don't suffer much, a few of them even get higher performances while using the MTL method, such as *Animal*, and *Charts*. That may benefit from the information from the samples of other high-level features.

Figure 3 also shows that the Bayesian networks improve the performance. Further work needs to be done on mining more on the relationships between these high-level features.

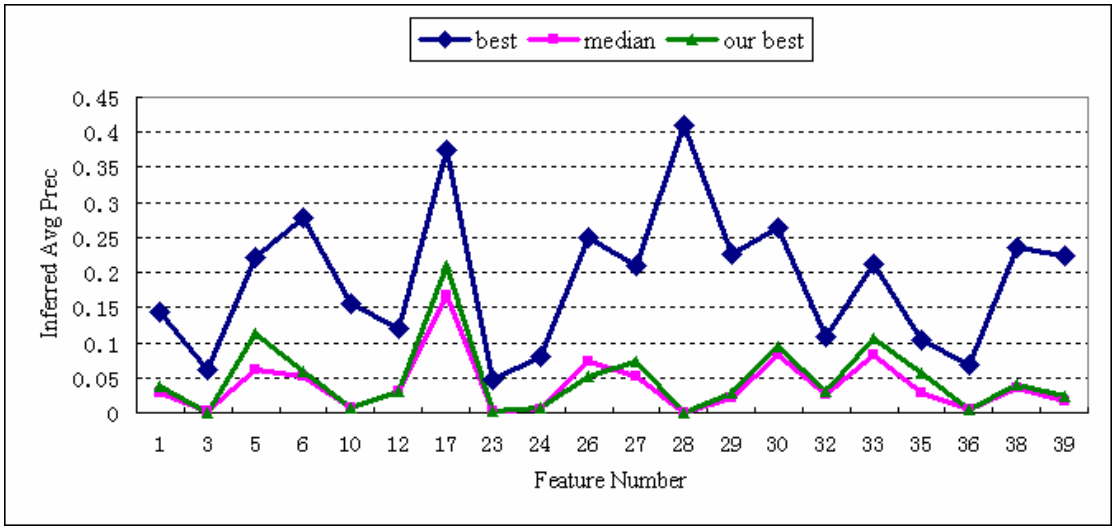

Figure 2 Comparison of our best results to other results

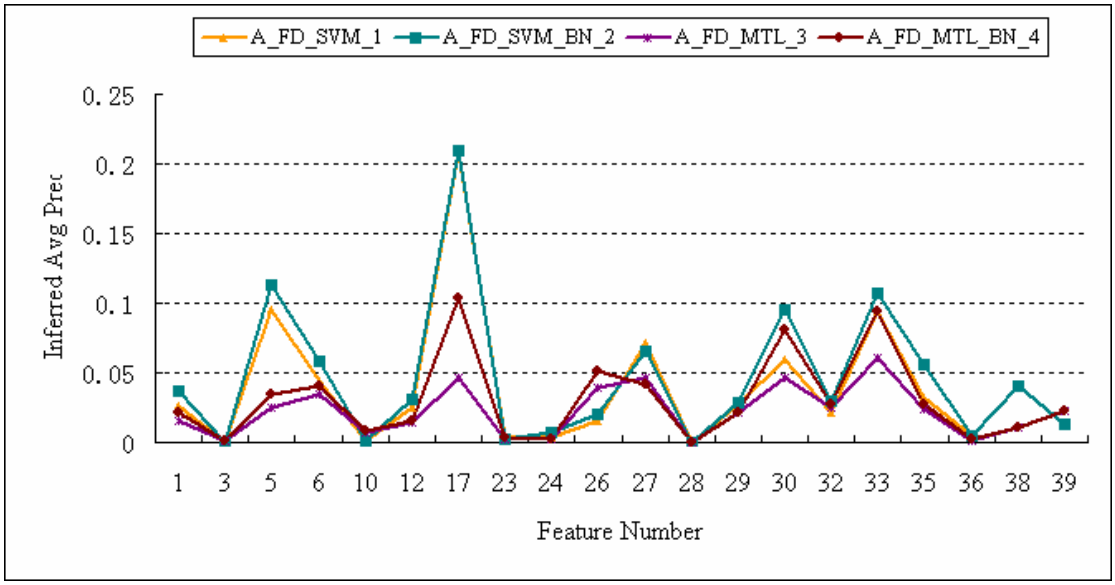

Figure 3 Comparison of our runs

3. Search

For the search task of TRECVID 2007, we submitted 5 interactive runs including 2 of them based only on text/image retrieval system alternatively as the baselines of our work. We continued to use multi-modal information fusion which performed well last year [2]. And the fusion are performed in different approaches. Methods such as machine learning and statistics were modified and optimized to extract more information as well as fit the framework of our interactive search scheme. In the following we will introduce these methods in detail.

3.1. Multi-Model 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 text description, global features, visual concepts and camera motion features. Every model only searches the video database using one kind of features which may not

be able to obtain satisfying query results. However, the multi-model fusion methods make good use of each model's particular characteristics and could always achieve better performance. Figure 4 illustrates the main architecture of our system.

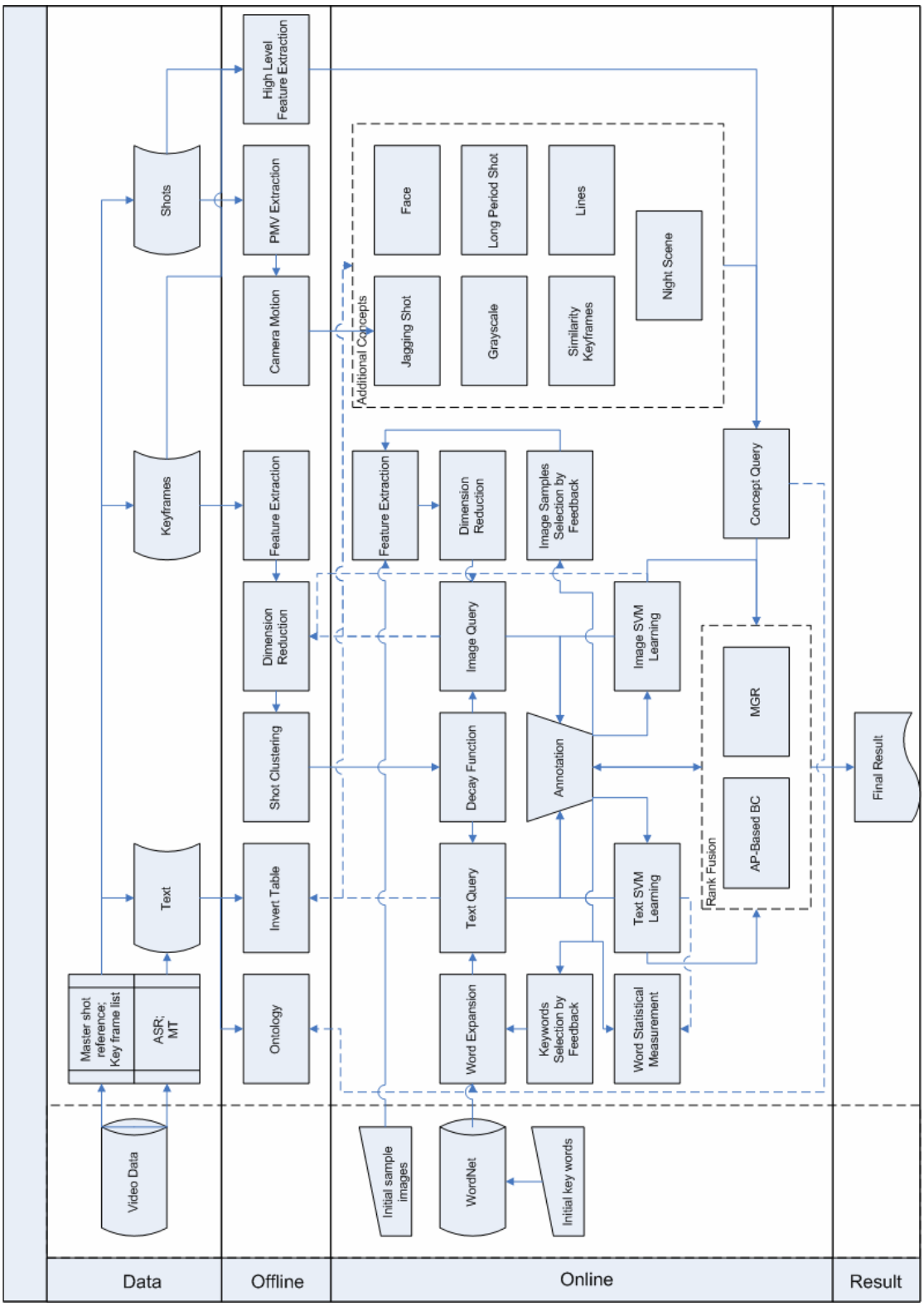

Figure 4 System architecture

3.1.1. Text Retrieval

The Text Retrieval Module is based on a Reverse Table which maps a word appeared somewhere in the video transcript (including ASR [3], MT results provided by NIST) to the shot IDs corresponding to those transcripts. This table is built in a nature way that words are distinguished totally by string comparison.

In order to perform a query, a keyword list is needed as the input to the system. The list comes manually in the initial time, and can be expanded by some specific tools or from the right samples by the method which we will refer to later. Our tool expands a query by listing the words nearby in the Reverse Table and the words in different tense. In addition, the WordNet [4] is also used. For the query results, the TF * IDF weighting scheme is adopted to generate the text retrieval scores of shots. The keywords in the list are weighted differently according to their positions in the sequence, i.e., the first word weights $c_1 = 1$, the second $c_2 = 0.9$, and so on, and the smallest weight is set to 0.5 in our experiment. Each shot mapped from a keyword by the Reverse Table is scored by the word's weights w_0, and its nearby shot is scored by a decay function w_i, where $w_i = w_0 \times (0.5)^i$, with i being the distance from the "hit" shot.

The Text Retrieval procedure is iterative and the keyword list is refined by the method, i.e. Keywords Training by Feedback (KTF), to enrich the query of each topic. The initial keyword list is formed manually by an intuitive sense, and the tools which have been inferred previously are used to refine the query. After the annotation of the scored shot list produced by the module, the words in the positive shots are collected and ranked by the frequency of their appearance in the positive shots. The system eliminates the words with too high or low frequency, which means they are unnecessarily popular or rare in the case, and add the remaining to the keyword list to form another query.

3.1.2. Image Retrieval

In our image retrieval module, four different global features, i.e., SCD (Scalable Color Descriptor), CLD (Color Layout Descriptor), CSD (Color Structure Descriptor), EHD (Edge Histogram Descriptor), all defined in MPEG-7, are extracted for each representative keyframe of the video shots after our search experiments, taking into account information from three kinds of color spaces (HSV, YUV, HMMD) and one kind of global texture.

For every query, several image examples are derived from Google [5] or development data by the criterion of NIST. Initial image features are extracted in either the example set or test data collection. After the extraction step, PCA (Principal Component Analysis) method is used for dimension reduction to speed up the query. The energy of PCA is 0.98 and finally we transform the initial feature vector from a 604-dimensional vector into a 78-dimensional vector, which is the final image feature used.

Similar to the text module, we refine the query examples during each iteration of the retrieval procedure. Initially, shots corresponding to the first 100 results in the ranked keyframe list are picked out for annotation. Since the results are usually not so satisfactory, we choose some of the positive results as the feedback image and add them to the example set, namely Query Examples

Training by Feedback (QETF). The method adopted for choosing the feedbacks is to use K-means clustering to cluster the positive results among the first several results into *n* clusters (the number *n* depends on the complexity of the topic) while the centroids of clusters are fed back to the query procedure. The recursion usually takes three or four times and the refined shot list can then be used in Machine Learning module or other modules.

In addition, successive shots often have similar content in the search task, so it is helpful to explore nearby shots of each positive result. We use the image feature in Image Retrieval Module to find such shots. The distances between the result and all its prior/posterior shots are calculated to produce a neighbor shot list according to some threshold value, which can be used in the decay function of the Text Retrieval Module.

3.1.3. Visual Concepts

Visual concepts are the high-level features which have been selected from one of the TRECVID tasks. This year we mainly used the results from Tsinghua University. Ontology is also introduced in our concept selection. By selecting relative semantic and logic to the query topics, the ontology can decide how many and which concepts are needed automatically.

In our ontology, all the concepts need to be mapped are divided into two parts: salient object and concepts. The salient object can be selected directly by keywords. Then the concepts reasoning in terms of ontology and salient objects of video are applied to map all the concepts.

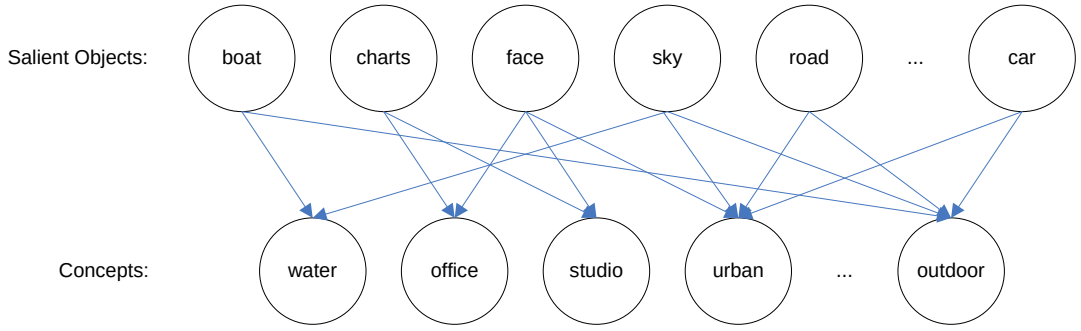

Figure 5 Ontology of concept detection

Figure 5 illustrates the ontology of concept detection, in which every two concepts in different layer has a line. The line presents the probability dependence relationship (weight) between the two concepts. We compute the weight by conditional probability. After the ontology has been constructed, below equation can be used to compute the probability of every concept.

$$P(c_k / S) = \sum_{i=1}^{N} (w_{ki} * P(s_i))$$

Where $P(s_i)$ is the confidence of the salient object, the set of salient object is $S = \{s_1, s_2, s_3, \ldots, s_M\}$, the set of all the high-level concepts is $C = \{c_1, c_2, c_3, \ldots, c_N\}$, and the weight of every c_k in C which corresponds to every salient object is $W_k = \{w_{k1}, w_{k2}, w_{k3}, \ldots w_{kM}\}$.

We judged a concept being in a test query or not by comparing the joint probability to a threshold. Hence all the related concepts can be used for a certain query. For instance, when we search the

topic “Find shots of a canal, river, or stream with some of both banks visible”, the ontology will select several concepts which are related to the scene described in the query, such as *boat_ship*, *water* and so on.

Besides, we introduced some complementary concepts to meet the need of searching particular topics. For example, the long-period shot concept by comparing the global feature of the frames to determine their similarities is used in the topics whose queries imply less movement, especially for some topics about interview, from which the frames during a specific time are always similar. Additionally, some other concepts are involved. We use the face concept in the topics mainly about people, the grayscale concept in the topics of some documentaries or historic scenes, night concept in topics with night information, etc. All these concepts were proved to be useful in our search task.

3.1.4. Camera Motion

In some videos, the estimation of camera motion can provide useful information to identify high-level events such as wide-angle and close-up views in sports videos. Consequently, in our system, we employ camera motion as one of the low-level features. In our camera motion retrieval subsystem, we mainly adopt the method proposed in [6] which estimate camera motion directly from MPEG video without full-frame decompression. First, MPEG motion vectors, indexed by the frame number, are extracted from every shot in the test data collection. Then the method provided by [6] is adopted to obtain the estimation of camera motion from the P-frames of MPEG motion vector. Let p1, p2, …, p6 denote the resultant estimates for six varieties of camera motion. Thus, we could pick out P_i as a low-level feature, to build up a specified camera motion (tilt or zoom) vector of each shot. Let $p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(n)}$ denote the vector, so we can get the camera motion information of the specified shot by using the following equation:

$$sum = \begin{cases} sum+1, & p_i^j > p_i^{j-1} \\ sum, & p_i^j = p_i^{j-1} \\ sum-1, & p_i^j < p_i^{j-1} \end{cases} \qquad j \in [0, n]$$

where *n* is the total number of P-frames in the shot, and *sum/n* denotes the level of the specified camera motion. The positive value of *sum/n* means tilting down or zooming in, and vice versa. The lager the absolute value of *sum/n* has, the more obvious the specified camera motion shows. Therefore, we can sort all the shots of the video of the test collection.

3.1.5. Cross System Retrieval

Like the KTF or QETF methods, the positive shots of the ranked lists from the text and image retrieval module could be selected by some annotation first. Additional text and visual information which may become new key words for the text module or new query examples for the image module is then extracted using the selected positive samples and fed to those retrieval modules correspondingly. As an intuitive text and image module fusion method, cross system retrieval could always increase the recalls of the system since more positive information is involved.

3.1.6. Machine Learning Methods

There are many different machine learning methods such as SVM, neural network, adaboost, etc. In our system we use SVM as the main classifier for its good performance in the search task. However, we have to face two problems in TRECVID: not enough positive samples and too much negative samples [7]. To solve this imbalance problem, K-Means clustering method is used to cluster the negative samples into several clusters, which makes the number of negative samples smaller. From the experiments, we find out that the search performs well when the cluster number is equal to the number of positive samples [8].

An SVM model with RBF kernel and a parameter *nu* to control the smoothness of the decision boundary is adopted to train an appropriate classifier [9]. Since there are too many varieties among different topics, we train specific classifier for each topic instead of using a uniform model. After the model is trained, it can be used to predict confidence values with positive or negative judgments of some incoming inputs for its corresponding topic, and output the ranked shot list.

3.1.7. Multi-Model Fusion

We use search methods which adopt multi-model queries respectively and merge the results through two different approaches: AP-Based Borda Count (BC) [10] and Mixed Group Ranks (MGR) [11]. As can be seen in the overall framework of our multi-modal video retrieval system from Figure 4, we take every module as an atom search engine and use them to search in video database respectively. According to these algorithms, different methods are then used to merge the query results and MC (Merge Confidence) is calculated to rank the final results. In AP-Based BC, MC is related to the precision of each atom search engine while in MGR it is refined from users' feedbacks and takes into account the confidence of each rank in the ranked list as well. From our experiments, the former could yield higher precision while the latter is able to find more positive shots, and both methods achieve better performance than other runs without fusion.

3.2. Experiments

The general procedure we adopted in each interactive run of the search task this year is as follows:

- work done off-line

Every task from data pre-processing to the construction of each retrieval system was finished off-line, as well as the ontology and high-level features.

- work in ready queue

In this part, what we do is to find a set of example shots and the query keywords for a given topic, which is formulated in multi-modalities including text description, global feature, visual concepts, and camera motion features.

- first-round annotation

The aim of the first-round annotation is to increase the recall of a certain query. Using KTF and QETF as described above or the cross retrieval method in the system, we could always find more right shots after such processes.

- second-round annotation

The aim of the second-round annotation is to increase the precision of the system, which mainly interacts with the training of image and text machine learning modules.

- third-round annotation

The last-round annotation helps to provide the information of each model's performance to the fusion module to train the specific weights for each topic used in combining the results.

3.2.1. Manual Browsing Strategy

Manual Browsing with resizing pages [12] is the annotation strategy we choose for interactive search, which gives the users more control on the display of the shots. It allows users to browse the shots ranked by computers with different sizes and played at different speeds. For the rate of recalls occurring in the sequence is not constant, there are usually more right shots in the top-rank shots than lower, which makes sense to use different page size between top-ranked shots and lower-ranked shots. Specifically, at the beginning stage, the shots are top-ranked by computer, which means they are more likely to be relevant and the frequency of relevant shots is high, we prefer to use smaller size of page, so that more attention can be paid on the shot of top-ranks. Later when the number of recalls decreases, the users may choose to use larger size of page. It can save time and make the operation more convenient. Instead of turning pages by system, users can click buttons to let the page forward or backward. Then it is less likely for users to miss the relevant shots.

- page sizes: The two choices about page size we give users to label shots are 1*1 and 2*2. At first we prefer to use 1*1, which can achieve more accurate results. Later, 2*2 is more time-efficient than 1*1, and labeling shots with 2*2 is more convenient. But there may be some difficulties if we use more than 4 shots, because inspecting 4 shots per page is hard for users and not necessarily efficient.
- precision and recall indications: The precision and number of positive shots of the topic which is being labeled is shown in the middle right of the screen, which gives a brief description about how the returned shots are relevant to the topic.
- manual paging: Although automatic paging is efficient for users to label shots, manual paging can avoid missing relevant shots. Since how long a user spends on the page is affected by many factors such as the page size, the complexity of the shots, the number of relevant shots, etc., we cannot set a rigid standard about the interval between two pages, otherwise users may sometimes need to wait for next page while sometimes they cannot catch up the showing of the shots. What's more, users also need to check whether they labeled correctly and manual paging is necessary. Compared with automatic paging, it can avoid some errors to let the users to turn pages by pushing buttons. And if we use keyboard to achieve these goals, it would be more convenient and time-efficient to users.

3.2.2. Evaluation

We submitted 5 interactive runs to the search task of TRECVID 2007 for evaluation. They are:

- FUDAN_P: based on multi-model fusion method using AP-Based BC.
- FUDAN_R: based on multi-model fusion method using MGR.
- FUDAN_C: based on cross-system retrieval.
- FUDAN_T: baseline run based on the text from the English ASR/MT output.
- FUDAN_I: baseline run based on image retrieval system.

Experimental results of the average precisions returned from NIST of our submissions against other submissions are shown in Figure 6. Besides, Figure 7 shows the comparisons of our runs.

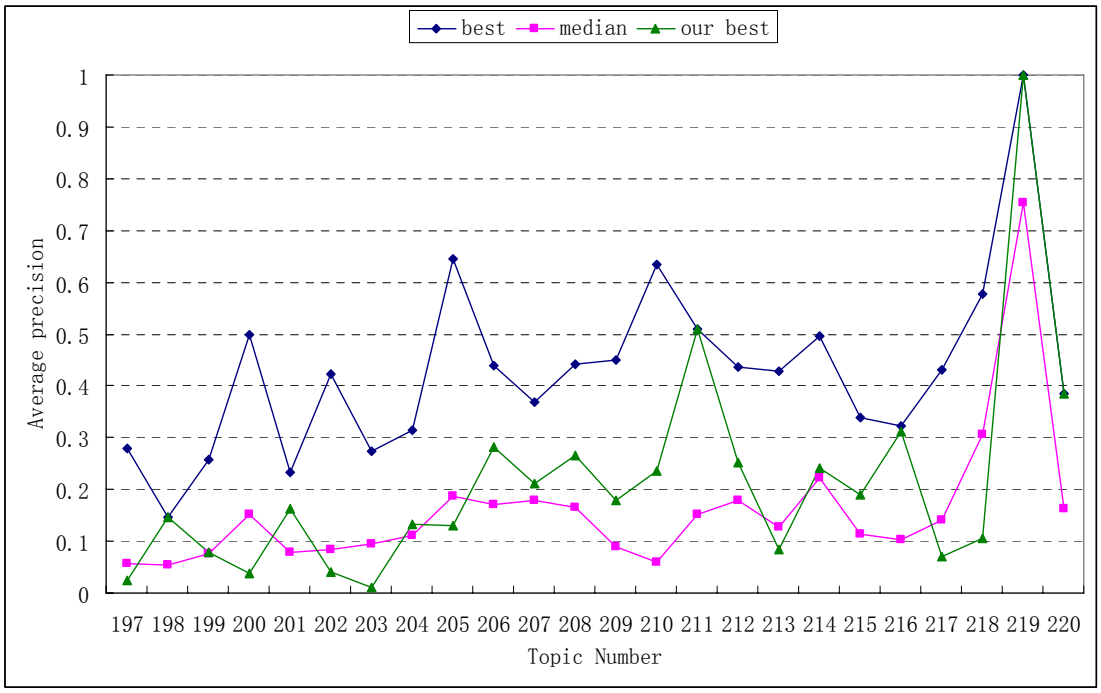

Figure 6 Comparison of our best results to other results

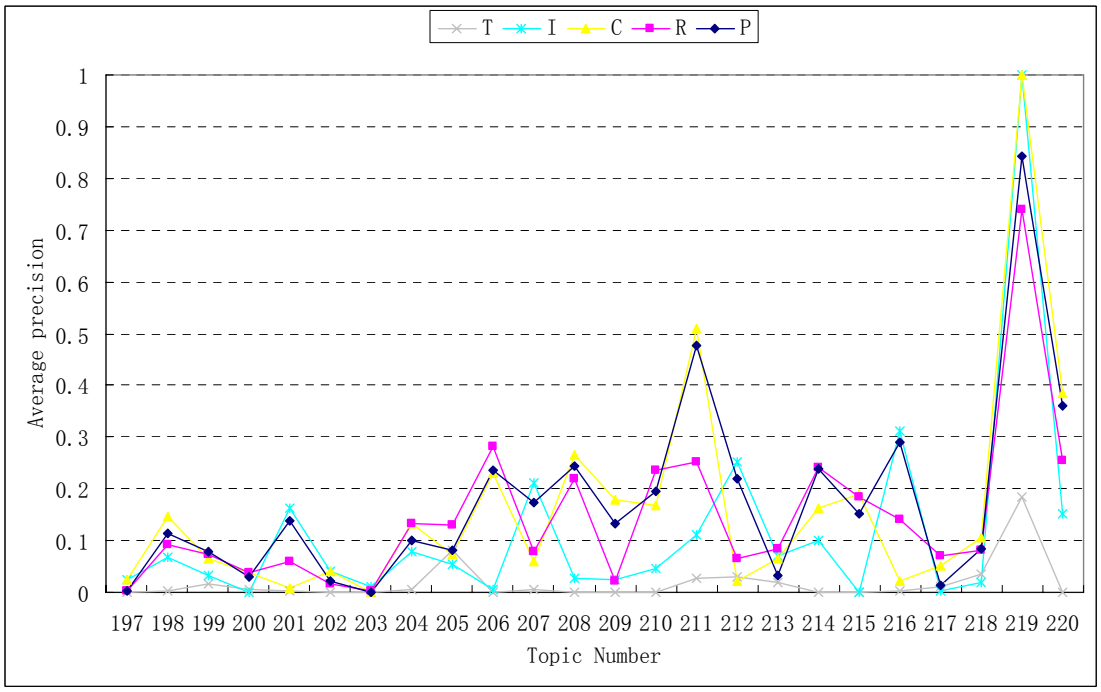

Figure 7 Comparison of our runs

It can be observed from Figure 6 that although 4 of our topics have equal or close performance with the best results from others, the remaining still needs further improvement, especially for those topics with poor results below median which involve complex semantic, such as *people walking upstairs*, *street market scene*, etc.

Among all our runs, as can be observed from Figure 7, FUDAN_T showed a low boundary performance due to the comparatively poor semantic information from the plain text. And between

two runs which use multi-model fusion methods, run FUDAN_P which uses AP-Based BC owns the highest precision of all while the other run FUDAN_R which uses MGR owns the highest recall. It illustrates that the multi-model fusion methods are very helpful in video retrieval.

In addition, the results on some topics by different methods vary much and on some topics vary little. For example, the precision of topic *door being opened* by using FUDAN_C and FUDAN_I are much higher than that by using FUDAN_T, while the topic of finding *the cook character in the Klokhuis series* have similar results for all our runs. It is clear that some topics which have obvious visual features, such as *sheep or goats*, have good results by all runs except the run with only textual information.

4. Summary

For the high-level feature extraction tasks, we extract several features at 3 different scales, mainly focusing on the texture feature. Besides regular SVM training and fusion with the same weight, we've also tried a multi-task learning method and Bayesian networks to improve the performances. The MTL method may solve the unbalance problem if with a suitable task clustering. Bayesian networks refine the results output by SVM or MTL methods and the relationships between high-level features may help to improve the performance.

For the search tasks, we adopted many kinds of information sources under the framework of multi-model video retrieval. The baseline runs with pure text and image retrieval give the low boundaries of the system's performance while the multi-model fusion still showed its effectiveness. From the results of two different fusion approaches, we could figure out that the fusion algorithm is flexible between precision and recall, which implies the possibilities of further modifications for better fitting into the search tasks to achieve higher results in the future.

Acknowledgement

This work was supported in part by MoE Foundation under contract 104075 and Shanghai Municipal R&D Foundation under contracts 05QMH1403, 065115017 and 06DZ15008.

Reference

[1] Kai Yu, Volker Tresp, and Anton Schwaighofer, "Learning Gaussian Processes from Multiple Tasks", in Proc. of the 22nd Int'l Conf. on Machine Learning, Bonn, Germany, 2005.

[2] Xiangyang Xue, Hong Lu, Hui Yu, Shile Zhang, Bin Li, Jing Zhang, Jie Ma, Bolan Su, Yuefei Guo, "Fudan University at TRECVID 2006", in Online Proc. of TRECVID 2006.

[3] Marijn Huijbregts, Roeland Ordelman and Franciska de Jong, "Annotation of Heterogeneous Multimedia Content Using Automatic Speech Recognition", in Proc. of SAMT, December 5-7 2007, Genova, Italy

[4] WordNet, http://wordnet.princeton.edu/.

[5] Google Image Search. http://images.google.com/.

[6] Tan Y-P, Saur D D, Kulkami S R, Ramadge P J, "Rapid Estimation of Camera Motion from Compressed Video with Application to Video Annotation", in IEEE Transactions on Circuits and Systems for Video Technology, Vol.10, No.1, Feb. 2000.

[7] Arnon Amir, Janne Argillandery, Murray Campbellz, Alexander Hauboldz, Giridharan Iyengar, Shahram Ebadollahiz, Feng Kangz, Milind R. Naphadez, Apostol (Paul) Natsevz, John R.

Smithz, Jelena Te˘si´cz, Timo Volkmer, “IBM Research TRECVID-2005 Video Retrieval System”, in Online Proc. of TRECVID 2005.

[8] J. Cao, Y. Lan, J. Li, Q. Li, X. Li, F. Lin, X. Liu, L. Luo, W. Peng, D. Wang, H. Wang, Z.Wang, Z. Xiang, J. Yuan, W. Zheng, B. Zhang, J. Zhang, L. Zhang, X. Zhang, “Intelligent Multimedia Group of Tsinghua University at TRECVID 2006”, in Online Proc. of TRECVID 2006.

[9] LibSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[10] Le Chen, Dayong Ding, Dong Wang, Fuzong Lin and Bo Zhang, “AP-based Borda Voting Method for Feature Extraction in TRECVID-2004”, in 27th European Conference on Information Retrieval, Santiago de Compostela, Spain, 2005

[11] Ofer Melnik, Yehuda Vardi, and Cun-Hui Zhang, “Mixed Group Ranks: Preference and Confidence in Classifier Combination”, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.26, No.8, Aug. 2004.

[12] A. G. Hauptmann, M. Christel, R. Concescu, J. Gao, Q. Jin, W.-H. Lin, J.-Y. Pan, S. M. Stevens, R. Yan, J. Yang, Y. Zhang, “CMU Informedia's TRECVID 2005 Skirmishes”, in Online Proc. of TRECVID 2005.