PKU-ICST at TRECVID 2009: High Level Feature Extraction and Search

Yuxin Peng, Zhiguo Yang, Lei Cao, Jian Yi, Ning Wan,

Yuan Feng, Xiaohua Zhai, En Shi and Hao Li

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We participate in two tasks of TRECVID 2009: high-level feature extraction (HLFE) and search. This paper presents our approaches and results in the two tasks. **In HLFE task**, we mainly focus on exploring the effective feature representation, data imbalance learning and fusion between different data sets. In feature representation, we adopt five basic visual features and six keypoint-based BoW features, and combine them to represent each keyframe image. In imbalance learning, we propose two methods for this problem: OnUm and concept category. In the fusion between different data sets, we use three different training sets: (1) TRECVID 2009 training data set (Tv09), (2) TRECVID 2005 training data set (Tv05), and (3) Flickr images. **In search task**, we participate in two types of search tasks: automatic search and manual search. We explore multimodal feature representation, which includes visual-based features, concept-based feature, audio features and face features. Based on these features, two retrieval methods are jointly adopted for search task: pair-wise similarity measure and learning-based ranking. We achieve the good results in both tasks. In HLFE task, official evaluation shows that our team ranks 2nd in type A and 1st in types C, a and c. In Search task, official evaluations show that our team rank 2nd in automatic search and 1st in manual search.

1 High Level Feature Extraction

In the HLFE task of TRECVID 2009, we participate in all 4 types of evaluation. The National institute of standards and technology (NIST) totally defines 4 types of runs according to the used training data: A, a, C, and c. Type-a runs only use non-SV TRECVID data, while type-A runs can use all TRECVID data (SV and non-SV). SV data refers to the data sets of TRECVID 2007, 2008 and 2009, because they are donated by Sound and Vision organization (SV). And TRECVID data in other years are called non-SV data. Both type-a and type-A runs are not allowed to use non-TRECVID training data (e.g., web images). Type-c runs can use any training data except SV data, while type-C runs have not any limitation in the training data. In our six submitted runs, the 1st run belongs to type C, the 2nd, 3rd and 4th runs belong to type A, the 5th run belongs to type c, and the 6th run belongs to

type a. They are described as follows:

- C-PKU-ICST-HLFE-1 (A-PKU-ICST-HLFE-3 + c-PKU-ICST-HLFE-5): weighted fusion of A-PKU-ICST-HLFE-3 and c-PKU-ICST-HLFE-5.
- A-PKU-ICST-HLFE-2 (A-PKU-ICST-HLFE-3 + a-PKU-ICST-HLFE-6): weighted fusion of A-PKU-ICST-HLFE-3 and a-PKU-ICST-HLFE-6.
- A-PKU-ICST-HLFE-3 (visual feature + O3U3 + Tv09 + audio feature + concept category): early fusion of five basic visual features and six keypoint-based BoW features, and audio features are used for a few related concepts. Training by O3U3 classifier on Tv09 data, and utilizing concept category.
- A-PKU-ICST-HLFE-4 (visual feature + O3U3 + Tv09): early fusion of five basic visual features and six keypoint-based BoW features, and trained by O3U3 classifier on Tv09 data.
- c-PKU-ICST-HLFE-5 (visual feature + O2U2 + Flickr + a-PKU-ICST-HLFE-6): early fusion of five basic visual features and six keypoint-based BoW features, trained by O2U2 classifier on Flickr data, utilizing concept category, and fused with a-PKU-ICST-HLFE-6.
- a-PKU-ICST-HLFE-6 (visual feature + O2U2 + Tv05): early fusion of five basic visual features and six keypoint-based BoW features, trained by O2U2 classifier on a subset of Tv05 data, and utilizing concept category.

The evaluation results of our 6 runs are shown in Table 1. Official evaluation shows: in type-A runs, our team ranks 2nd in all 41 teams that submitted type-A runs (our best run ranks 4th among all 202 type-A runs of 41 teams, and the first three runs belong to the same team). In types C, a, and c runs, all our runs rank 1st.

Table 1: Results of our submitted 6 runs on HLFE task of TRECVID 2009.

ID	MAP	Brief description
C-PKU-ICST-HLFE-1	**0.205**	A-PKU-ICST-HLFE-3+ c-PKU-ICST-HLFE-5
A-PKU-ICST-HLFE-2	**0.203**	A-PKU-ICST-HLFE-3+ a-PKU-ICST-HLFE-6
A-PKU-ICST-HLFE-3	0.199	Visual feature+O3U3+Tv09+audio feature+concept category
A-PKU-ICST-HLFE-4	0.198	Visual feature+O3U3+Tv09
c-PKU-ICST-HLFE-5	**0.120**	Visual feature+O2U2+Flickr+concept category +a-PKU-ICST-HLFE-6
a-PKU-ICST-HLFE-6	**0.092**	Visual feature+O2U2+Tv05+concept category

The framework of our HLFE system is shown in Figure 1. Besides the training data set from TRECVID 2009(Tv09), we also use two other data sets: the TRECVID 2005 training data set(Tv05), and the web images downloaded from Flickr website (Flickr). For each of three training sets: (1) the same visual features are extracted, (2) the same OnUm algorithm is adopted (with different parameters) to handle the data imbalance problem, and (3) the same concept category method is used to exploit the inter-concept correlation. Audio features are only used in the TRECVID 2009 training data set. For the test set, five keyframes are uniformly extracted from each subshot and the same visual features are extracted. As shown in Figure 1, the six submitted runs are the separate or combined results of the three training data sets.

Our sixth run a-PKU-ICST-HLFE-6(Run6) only uses Tv05 data, while Run5 combines Tv05

data and Flickr data, and gains a big performance improvement over Run6. Run4, our baseline type-A run, without using the concept category method, already performs much better than Run6 and Run5, because both training data of Run4 and test data set are from the TRECVID 2009 data sets and have similar video content. In Run3, the usage of audio features and inter-concept correlation only has a slight improvement compared to Run4. Run2 combines the result of Run3 and Run6, while Run1 combines the result of Run3 and Run5, both gaining considerable increases over the separate results. This shows that the three training data sets are complementary for HLFE task.

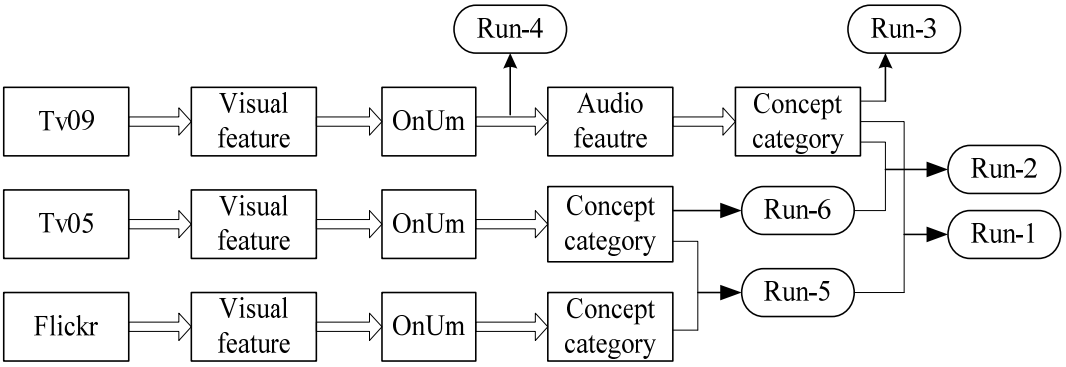

Figure 1: Framework of our HLFE approach for the submitted six runs.

1.1 Feature Representation

We use three kinds of features for the HLFE tasks, namely basic visual features, keypoint-based BoW features, and audio features. The basic visual features and keypoint-based BoW features are used for all 20 concepts, while the audio features are only used for three related concepts on *Person-playing-a-musical-instrument*, *Female-human-face-closeup*, and *Singing*.

1.1.1 Basic visual features

We extract five basic visual features namely CMG(Color Moment Grid), LBP(Local Binary Pattern), Gabor(Gabor wavelet texture), EHL(Edge Histogram Layout) and EOH(Edge Orientation Histogram) from each keyframe image. The details of these visual features are given as follows:

(1) **CMG** (225-d): the image is divided into sub-images by a 5x5 grid in the CIE-Lab color space. The color moments of the 1st, 2nd and 3rd orders are extracted from these sub-images in each channel.

(2) **LBP** (531-d): it depicts the relationship of the center pixel and P equally spaced pixels on a circle of radius R in a gray-scale image. We first divide the gray-scale image into sub-image by a 3x3 grid, and then choose a neighborhood size of 8(P = 8) equally spaced pixels on a circle of radius 1(R = 1) that form a circularly symmetric neighbor set with "uniform" patterns .

(3) **Gabor wavelet texture** (240-d): we first partition the gray-scale image into five regions, and then generate 24 Gabor filters in each region. The mean and standard deviation are computed by 24 Gabor filters over the 5 regions.

(4) **EHL** (320-d): We first partition the gray-scale image into five regions, and then we extract

edge histogram with 8 direction bins and 8 magnitude bins in each region.

(5) **EOH** (657-d): We first divide the gray-scale image into sub-images by a 3x3 grid, and then we extract edge points in each grid. A 73-bin histogram is computed for each region: the first 72 bins are used to represent edge pixels by their different directions, and the last bin is the number of non-edge pixels.

1.1.2 Keypoint-based BoW features

As in last year, we continue to explore the keypoint-based BoW(Bag-of-Word) features to represent each keyframe image. In our method, the extraction of keypoint-basd BoW features includes three steps:

(1) Detect keypoints from the images, and use SIFT descriptor[1] to extract 128-d feature vectors for the keypoints;

(2) Use k-means algorithm to cluster the keypoints into 500 clusters, and form a visual vocabulary with the cluster centroids;

(3) Adopt soft-weighting[5] method to assign keypoints to multiple nearest visual words(centroids), where the word weights are determined by keypoint-to-word similarity. The normalized histogram of visual words forms a BoW feature vector.

To improve the performance of BoW feature, in the step (1) we adopt six complementary detectors to detect the keypoints from images: Difference of Gaussian (DoG) [1], Laplace of Gaussian(LoG)[1], Harris Laplace[2], Dense sampling[9], Hessian Affine [3], and MSER [4]. For each detector, a 500-d feature vector is generated separately and six feature vectors are concatenated to form a 3000-d BoW feature, as shown in Figure 2. After that, we further combine it with the basic visual features in an “early fusion” manner, resulting in a 4973-d visual feature.

Figure 2: Combination of six keypoint-based BoW features.

1.1.3 Audio Features

We adopt two audio features namely NMF(Nonnegative matrix factorization)[6] and MFCC for three concepts that are closely related with sound, including *Person-playing-a-musical-instrument*, *Female-human-face-closeup*, and *Singing*. In NMF features, we use a 168-d vector [6]. In MFCC features, we use the first 29 coefficients and the log energy coefficient, concatenated with the delta and delta-delta coefficients of this vector, which is 90-d in total. These two audio features are 258-d in total.

1.2 Data Imbalance Learning

The data imbalance problem, which means the number of negative samples is far more than that of positive samples, is a major factor to affect the performance of classifiers. In TRECVID 2009, the *NPR* (ratio of negative samples versus positive samples) on the 20 concepts of HLFE task is shown in Figure 3. The NPR value varies from 27 to 680, and most concepts have NPR values larger than 100. In TRECVID 2009, the average value of NPR is 123, while this number is 93 in TRECVID 2008. The effective yet efficient methods are needed to solve the data imbalance problem.

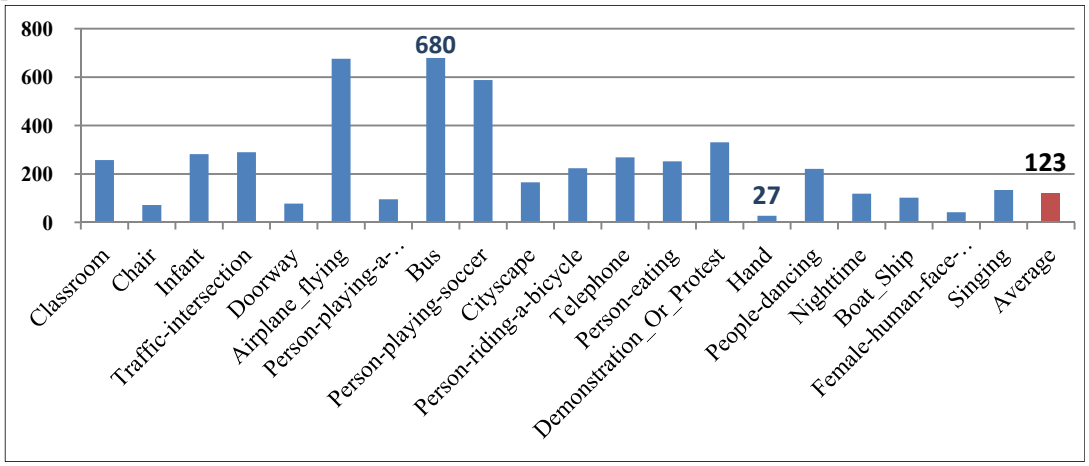

Figure 3: NPR on 20 Concepts in HLFE task of TRECVID 2009.

1.2.1 OnUm

This year we adopt OnUm method to handles the data imbalance problem for the effectiveness and efficiency, which mainly include the following steps(as shown in Figure 4):

(1) **Oversampling**: duplicate the original positive sample set P for $(n-1)$ times, and get a new positive sample set P', which contains n times positive samples as P.

(2) **Undersampling**: split the original negative sample set N into m parts $\{N'_1, N'_2, \dots, N'_m\}$ and combine each part with P' to get m subsets $\{E'_1, E'_2, \dots, E'_m\}$, where $\{E'_i = N'_i + P'\}$.

(3) **Combination**: Use each subset E'_i to train a model and predict on the test data set. In this step, any learning method can be used.

In our approach, we adopt *LibSVM* with *RBF* kernel and default parameters. Then the m prediction scores are averaged to produce the final score. The parameters in OnUm algorithm

$\{n, m\}$ are set according to the degree of data imbalance. More imbalanced data set needs bigger n and m. In our approach, for Tv09 data, we adopt O3U3 ($n = 3, \mathrm{m} = 3$); for Tv05 and Flickr data, we use O2U2 ($n = 2, \mathrm{m} = 2$). Experiments show our OnUm algorithm can achieve the effective and efficient results.

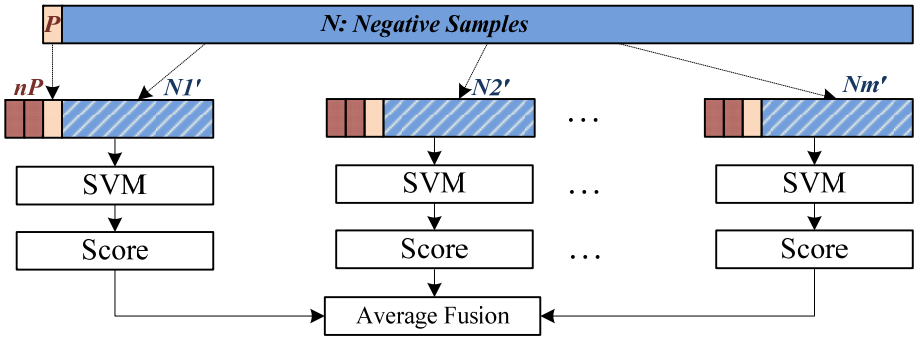

Figure 4: Diagram of OnUm algorithm.

1.2.2 Concept category

We further employ the concept category method to handle the data imbalance problem. In video data, each concept does not exist alone, but has the relationship with other concepts at the semantic level. Under this situation, we can classify the correlated concepts into a category, and use the union of the positive sample sets of the concepts in a category as the positive sample set of this category, which increases the number of positive examples. In our method, we manually classify 12 of the 20 concepts in the HLFE task into 4 categories according to their inter-concept correlation, as shown in Table 2. The positive samples of category $K \in \{\mathrm{AB}, \mathrm{CCT}, \mathrm{SPP}, \mathrm{BCTP}\}$ is the union of positive sample sets of all concepts that belong to K, that is, $\mathrm{P_K} = \mathrm{U_{c \in K}\, P_c}$, while the negative samples of category K is the union of negative sample sets excluding $\mathrm{P_K}$, that is, $\mathrm{N_K} = (\mathrm{U_{c \in K}\, N_c}) - \mathrm{P_K}$.

Table 2: Four concept categories.

Category	Concepts
AB	Airplane_flying, and Boat_Ship.
CCT	Chair, Classroom, and Telephone.
SPP	Singing, People-dancing, and Person-playing-a-musical-instrument.
BCTP	Bus, Cityscape, Traffic-intersection, and Person-riding-a-bicycle.

We adopt the following steps to calculate $prob(kf, c)$, that is, the probability score that keyframe *kf* contains concept *c*: (1) Get the original prediction score $prob\_orig(kf, c)$: use the positive samples($\mathrm{P_c}$) and negative samples($\mathrm{N_c}$) of concept *c*, to get the original prediction score with the learning method described in Section 3. (2) Get the prediction score of the concept category $prob(kf, K)$: use the positive samples ($\mathrm{P_K}$) and negative samples($\mathrm{N_K}$) of concept category *K*, to get $prob(kf, K)$ with the same learning method. $prob(kf, K)$ is the probability that keyframe *kf* contains at least one concept that belongs to the concept category *K*. (3) Get the final prediction score: $prob(kf, c) = prob\_orig(kf, c) \times prob(kf, K)$.

Since the concept category *K* has more positive samples than any concept in it, $prob(kf, K)$ is expected to have higher accuracy than $prob\_orig(kf, c)$ and can be used as a filter. By multiplying $prob\_orig(kf, c)$ with $prob(kf, K)$, the performance can be enhanced.

1.3 Fusion between different training data sets

This year, we totally use three different training data sets: Tv09, Tv05 and Flickr. For Tv05 data, we adopt the annotation data of 14 concepts from LSCOM; for the rest 6 concepts that are not included in LSCOM lexicon, we manually label the TRECVID 2005 training data. For the Flickr data, we use the concept names as keywords and search for images on the Flickr website. For each concept, the top 1000 images returned by Flickr are used as positive samples. Totally, 20,000 images are used. For each concept, all 1000 images belonging to the concept are used as the positive samples, and the 19000 images for other concepts are used as negative samples. So in the Flickr data set, the *NPR* value is always 19 for each concept. The Flickr images are only used in combination with Tv05 or Tv09 data. We do not use it alone for the following two reasons: (1) The Flickr images and the test data set are in very different domains; (2) The Flickr images generally contain noises.

In our six submitted runs, Run6 only uses Tv05 data. Run5 combines Flickr data with Tv05 data, and gains significant increase over Run6. Run4 and Run3 only use Tv09 data. Run2 is the weighted fusion of Run3 and Run6, while Run1 combines the result of Run3 and Run5, both gaining considerable performance increases over the separate results. Official evaluations show that the three training data sets are complementary and their fusion can improve the performance.

2 Search

In search task of TRECVID 2009, we participate in two of the three types: automatic search and manual search. We submitted 10 runs for the search task of TRECVID 2009, including 8 runs(6 normal + 2 high-precision) for automatic search, and 2 runs(1 normal + 1 high-precision) for the manual search. The evaluation results of our 10 runs are shown in Table 1. In automatic search, our team ranks 2nd in all 12 teams (our best run ranks 3rd among all 88 runs of 12 teams, and the first two runs belong to the same team) for the normal type, and achieve the best result among all six runs for high-precision type. In manual search, our run rank 1st for the normal type and we are the only team that submitted high-precision manual runs.

Table 3: Performance of our submitted 10 search runs.

Type	Condition	ID	MAP	Brief description
Automatic search	Normal	F_A_N_PKU-ICST-4_4	**0.098**	V+P+CS+F+A
		F_A_N_PKU-ICST-5_5	0.095	P+CS+F
		F_A_N_PKU-ICST-7_7	0.096	P1+CS+F
		F_A_N_PKU-ICST-8_8	0.080	CS
		F_A_N_PKU-ICST-9_9	0.095	P1
		F_A_N_PKU-ICST-10_10	0.090	P
	High-precision	F_A_P_PKU-ICST-3_3	0.236	V+P+CS+F+A
		F_A_P_PKU-ICST-6_6	**0.263**	P1+CS+F
Manual search	Normal	M_A_N_PKU-ICST-2_2	**0.126**	P+CSman+F
	High-precision	M_A_P_PKU-ICST-1_1	**0.354**	P+CSman+F

The brief description in Table 3 is based on our basic methods in Table 4. Besides, in Table 3, “A” means using audio features for the topics that are closely related with sound, and “F” means

re-ranking the result based on face detector. The framework of our system for search task of TRECVID 2009 is shown in Figure 5.

Table 4: Description of our methods

Our method	Description
Vmb	Retrieval by MBSVM(PCopy=1, NPR=5, BagNum=5) based on visual features, which extracts 5 frames from each video clip, and uses 5 copies of web images in query examples.
Vcos	Retrieval by cosine distance based on visual features, which extracts 5 frames from each video clip, and uses 1 copy of web images in query examples.
V	Weighted fusion of Vmb and Vcos.
Pmb	Retrieval by MBSVM(PCopy=100, NPR=5, BagNum=5) based on concept-based probability feature, which extracts 1 frame from each video clip, and uses 1 copy of web images in query examples.
Pcos	Retrieval by cosine distance based on concept-based probability feature, which extracts 5 frames from each clip, and uses 1 copy of web images in query examples.
P	Weighted fusion of Pmb and Pcos.
Pmb1	Retrieval by MBSVM(PCopy=1, NPR=5, BagNum=5) based on concept-based probability feature, which extracts 1 frame from each video clip, and uses 1 copy of web images in query examples.
Pcos1	Retrieval by cosine distance based on concept-based probability feature, which extracts 1 frame from each clip, and uses 1 copy of web images in query examples.
P1	Weighted fusion of Pmb1 and Pcos1.
CS	Retrieval based on automatic concept selection (for automatic search).
CSman	Retrieval based on manual concept selection (for manual search).

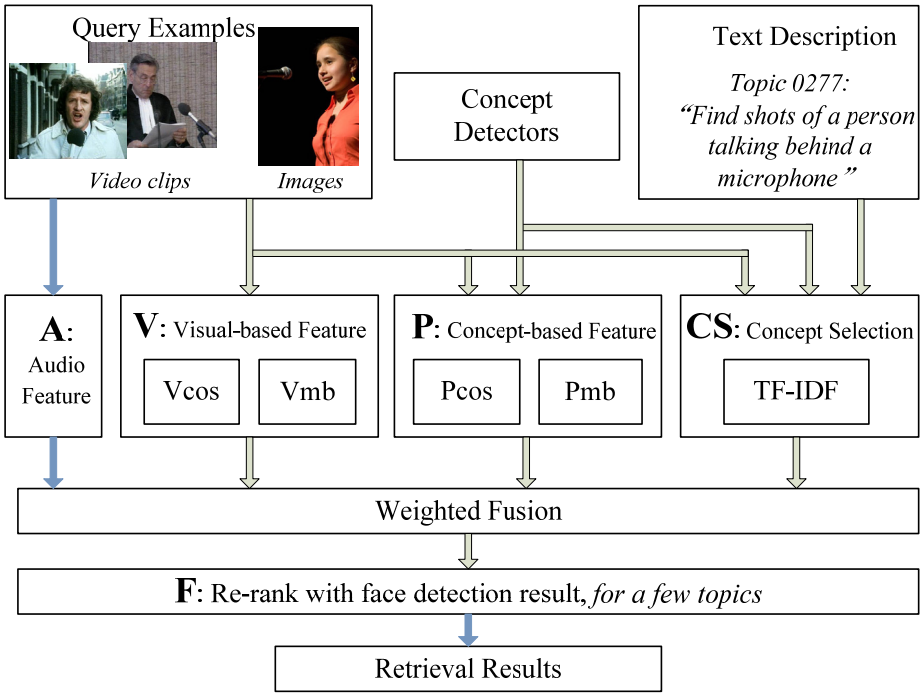

Figure 5: Framework of our search system.

2.1 Feature Representation

In search task, we use four kinds of features: visual-based features, concept-based features, audio features, and face features. The visual-based features and concept-based features are jointly used for all topics, while the audio features and face features are only used for a few related topics. In visual-based features, we employ the same features as in our HLFE system of TRECVID 2009, that is, the combination of five basic visual features and six keypoint-based BoW features (see Section 1.1.1 and 1.1.2), which is a 4973-d feature vector for each keyframe image. In addition, we use 117-d concept-based probability feature for the search task, which is extracted as follows: (1) A 117-concept lexicon is generated as follows: (a) All the concepts in the HLFE task of TRECVID 2007, 2008 and 2009 are selected. There are totally 65 such concepts. (b) 52 related concepts in LSCOM [8] lexicon are also used. (2) For each keyframe image, these 117 concepts models based on our HLFE system of TRECVID 2009 produce 117 prediction scores, which form a 117-d probability feature vector. Due to the large number of concepts and limitation of computing resources, we adopt "late fusion" instead of "early fusion" among the visual-based features. Each feature is used separately with OnUm algorithm to produce a prediction score for each keyframe. Then the 11 prediction scores given by 11 visual features are averaged to produce the final prediction score. In audio features, we use the same 258-d audio features as in our HLFE task (see Section 1.1.3). In addition, we also adopt face feature for the topics that are closely related with human face, and the details are shown in Section 2.4.

2.2 Retrieval method

We employ jointly two methods for retrieving the relevant shots with the query topics. That is, pair-wise similarity measure and learning-based ranking.

2.2.1 Pair-wise similarity measure

We employ the cosine distance to measure the similarity value between a query topic and the test shot in the data set, which is described in Figure 6.

2.2.2 Learning-based ranking

We also adopt learning-based method to rank relevant shots for a given topic. The query examples, including the images and the frames extracted from video clips, are considered as positive samples. Due to the fact that only a very small part of the shots are relevant with the topics in the test data set, we adopt the test data as negative examples. A problem based on learning-based method is that there are too few positive samples and too many negative samples, and the data imbalance problem is more serious than that in the HLFE task. In our approach, we use MBSVM algorithm to handle this problem, and the details are presented in Figure 7.

(1) Calculate the similarity(distance) value between each *e* and each *f*, where *e* is a query example of topic *T*, that is, a web image or a frame extracted from a video clip, and *f* is an extracted frame(5 frames per subshot) in test data set. The similarity value between *e* and *f* is denoted as $Sim(e, f)$.

(2) For each *f*, calculate the average of its similarity values with all query examples, as its similarity value with topic *T*:

$$Sim(T, f) = \sum Sim(e, f)/|T|$$

where $e \in T$ is a query example of topic *T,* and $|T|$ is the number of query examples.

(3) For each shot *s* in test data set, average the similarity values of its all extracted frames with topic *T*, as its similarity value with topic *T*:

$$Sim(T, s) = \sum Sim(T, f)/|s|$$

where $f \in s$ is a frame extracted from a subshot in shot *s,* and $|s|$ is the number of extracted frames in shot *s*.

Figure 6: our algorithm for pair-wise similarity measure.

(1) Over-sample the positive samples: Duplicate the positive sample set P for $(PCopy - 1)$ times and get a new set of positive samples P' with $PCopy \times PN$ samples, where *PN* is the number of positive samples in P before over-sampling.

(2) Under-sample the negative samples: Randomly select $R \times PCopy \times PN$ negative samples, and combine them with the over-sampled positive sample set P' to form a bag. That is to say, in each bag, the number of negative samples is *R* times as the number of positive samples, where *R* is a parameter to control the degree of data imbalance in each bag. A model is trained by *LibSVM* for each a bag, where *RBF* kernel is used with default parameters.

(3) Repeat the above step (2) for *BagNum* times, where *BagNum* is a parameter specifying the number of bags. Then for each shot in the test data set, the *BagNum* prediction scores given by different models are averaged to form the final result. Notice that the negative samples in each bag are selected without repetition, that is, the negative samples in these bags are totally different. This ensures that we can make full use of most of negative samples.

Figure 7: our algorithm for learning-based ranking.

Totally, there are three important parameters in MBSVM algorithm: *PCopy*, *R* and *BagNum*. Experiments show that *R=5* and *BagNum*=5 can achieve good performance in both the accuracy and efficiency, while *PCopy* needs to be set according to the number of frames extracted from each video clip in the query examples. Compared with OnUm methods in our HLFE task, MBSVM algorithm is more suitable for the search task because the data set is more imbalanced.

2.3 Concept selection

The search task based on concept selection can be divided into two steps: (1) Select the relevant concepts for each topic, and assign proper weight to each of the selected concepts; (2) Use the weighted average of the prediction scores of selected concepts to measure the similarity between the test shots and the topics. In step (1), the relevant concepts can be selected both automatically

and manually. For automatic concept selection, the concepts are selected according to the query examples, which is our approach for automatic search task. The details are given in Figure 8. For manual concept selection, the concepts are selected by human according to the text description and query examples of the topics, which is our approach for manual search task.

(1) For each topic T, calculate the average prediction score of the query examples for each concept. If topic T contains M images and N video clip examples, we extract the middle frame from each video clip, and get $M+N$ image examples. For concept C, we calculate the average value of the prediction scores of these $M+N$ image examples, which is denoted as $ExampleScore(T,C)$. Larger $ExampleScore\,(T,C)$ value means greater correlation between topic T and concept C.

(2) For each concept, we calculate the average value of prediction scores on the test shots, which is denoted as $TestScore(C)$. Larger $TestScore(C)$ value means concept C is more common and less discriminative.

(3) The final relevance value between topic *T* and concept C is calculated as follows:

$$Relevance(T,C) = ExampleScore(T,C) \times \log_2 \frac{1 - TestScore(C)}{TestScore(C)}$$

This formula derives from the *TF-IDF* formula in the field of text retrieval, where $ExampleScore(T,C)$ is similar to *TF*, and $\log_2 \frac{1-TestScore(C)}{TestScore(C)}$ is similar to *IDF*.

(4) For each topic, the three concepts with the highest $Relevance(C,T)$ values are selected, which are denoted as $S(T) = \{C1, C2, C3\}$.

(5) Searching all of the concepts in order to expand the three concepts selected for topic *T*. The concepts in the lexicon set *L* are divided into 16 categories (categories with only one concept are not counted). Add concept *C'* in *L* into the set of relevant concepts $S(T)$ if it satisfies the following two conditions: (I). *C'* falls into the same category with one of $\{C1, C2, C3\}$, and (II). *C'* has a $Relevance(T,C)$ value that ranks top 30 in *L*.

(6) Remove stop concepts from $S(T)$. The concepts *sky* and *outdoor* are considered as stop concepts, which are similar to stop words in the field of text retrieval. These two concepts are removed from the set of relevant concepts $S(T)$.

Figure 8: our algorithm for concept selection.

2.4 Re-ranking with face detection

We use the face detector to re-rank the retrieval results for the topics that are closely related with human face. We adopt a cascade face detector [7] to detect faces in the test shots. With the face detection result, different re-ranking methods are adopted for two topics: For **topic 0277**, that is, "*find shots of a person talking behind a microphone*", we use the following re-ranking method: (1) Divide the test shots into two types: the shots with human face, and the shots without human face. (2) All the shots with human face are re-ranked before the shots without human face, while the order of the shots with human face and the order of the shots without human face are kept unchanged. For **topic 0292**, that is, "*find shots with the camera zooming in on a person's face*", a shot is selected into the front sub-list only when: (1) All of the five keyframes that are uniformly positioned in the shot contain exactly one human face; and (2) The face sizes in the five keyframes keep increasing.

3 Conclusion

By participating in the HLFE task in TRECVID 2009, we have the following conclusions: (1) Effective feature representation is still vital, (2) The imbalance data learning is a key factor, (3) The fusion among different training data sets can improve the performance.

By participating in the search task of TRECVID 2009, we have the following conclusions: (1) Effective visual-based features and concept-based probability features provide the strong basics, (2) Learning-based retrieval is more effective than pair-wise similarity measure, (3) In concept selection, weighted fusion of the prediction scores of relevant concepts can give good result for the search task, and (4) audio features and face detector are useful for the topics with the distinguishing audio and face features.

Acknowledgements

The work described in this paper was fully supported by the National Natural Science Foundation of China under Grant No. 60873154 and 60503062, the Beijing Natural Science Foundation of China under Grant No. 4082015, and the Program for New Century Excellent Talents in University under Grant No. NCET-06-0009.

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision(IJCV)*, vol. 60, no.2, pp.91-110, 2004.

[2] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: A survey”, *Foundations and Trends in Computer Graphics and Vision*, vol.3, no.3, pp. 177-280, 2008.

[3] K. Mikolajczyk, C. Schmid, “Scale and affine invariant interest point detectors”, *International Journal of Computer Vision(IJCV)*, vol. 60, no. 1, pp. 63-86, 2004.

[4] J. Matas, O. Chum, M. Urban, T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, *British Machine Vision Conference(BMVC)*, pp.384-393, 2002.

[5] Y.-G. Jiang, C.-W. Ngo, J. Yang, “Towards optimal bag-of-features for object categorization and semantic video retrieval”, *ACM international conference on Image and video retrieval(CIVR),* pp.494-501, 2007.

[6] André Holzapfel, Yannis Stylianou, “Musical Genre Classification Using Nonnegative Matrix Factorization-Based Features”, *IEEE Transactions on Audio, Speech, and Language Processing(TASLP)*, vol. 16, no. 2, pp.424-434, 2008.

[7] J. Wu, S. C. Brubaker, M. D. Mullin and J. M. Rehg. “Fast Asymmetric Learning for Cascade Face Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 30, no. 3, pp. 369-382, 2008.

[8] LSCOM Lexicon Definitions and Annotations Version 1.0, DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, Columbia University ADVENT Technical Report #217-2006-3, Mar. 2006.

[9] L. Fei-Fei and P. Perona, “A bayesian hierarchical model for learning natural scence categories”, *CVPR*, vol.2, pp. 524-531, 2005.