Florida International University and University of Miami TRECVID 2011

Chao Chen, Dianting Liu, Qiusha Zhu, Tao Meng, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33146, USA
{c.chen15, d.liu4, q.zhu2, t.meng}@umiami.edu, shyu@miami.edu

Yimin Yang, HsinYu Ha, Fausto Fleites, Shu-Ching Chen
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
{yyang010, hha001, fflei001, chens}@cs.fiu.edu

Winnie Chen, Tiffany Chen
Miami Palmetto Senior High School, Pinecrest, FL 33156, USA

Abstract

This paper presents a summary of the team "Florida International University - University of Miami (FIU-UM)" in TRECVID 2011 tasks [1]. This year, the FIU-UM team participated in the Semantic Indexing (SIN) and Instance Search (INS) tasks. Four runs of the SIN results were submitted.

- F_A_FIU-UM-1_1*: KF+Meta&Relation+Audio+SPCPE&SIFT - Fuse the results from Subspace Modeling and Ranking (SMR) using the Key Frame-based low-level features (KF), LibSVM classification using Metadata from those meta-xml files associated with the IACC videos as well as the relationships between semantic concepts (Meta&Relation), Gaussian Mixture Models (GMM) using the Mel-frequency cepstral coefficients (MFCC) audio features, and the simultaneous partition and class parameter estimation (SPCPE) algorithm with scale-invariant feature transform (SIFT) interesting points matching (SPCPE&SIFT).*
- F_A_FIU-UM-2_2*: KF+Meta&Relation - Fuse the results from SMR using KF and LibSVM using meta information and relationships between semantic concepts.*
- F_A_FIU-UM-3_3*: KF+Audio+SPCPE&SIFT - Fuse the results from SMR using KF, GMM using MFCC audio features as well as SPCPE&SIFT matching.*
- F_A_FIU-UM-4_4*: KF - Served as the baseline model which uses SMR on the Key Frame-based low-level features.*

In addition, four runs of the INS task were also submitted.

- *FIU-UM-1: Use* 95 *original example images as well as* 261 *self-collected images to train the Multiple Correspondence Analysis (MCA) models to rank the testing video clips according to each image query; and use SIFT, K-Nearest Neighbor (KNN), and related SIN models to re-rank the returned video clips.*
- *FIU-UM-2: Use* 95 *original example images as well as* 261 *self-collected images to train the MCA models to rank the testing video clips according to each image query; and use the MCA model trained by* 95 *original example images to re-rank the video clips.*

- *FIU-UM-3: Use* 95 *original example images as well as* 261 *self-collected images to train the MCA models to rank the testing video clips according to each image query; and no re-ranking processing is performed in this run.*
- *FIU-UM-4: Use* 95 *original example images to train the MCA models to rank the testing video clips according to each image query; and re-rank the testing video clips by the KNN results obtained by* 95 *original example images.*

After analyzing this year's results, a few future directions are proposed to improve the current framework.

1 Introduction

The semantic indexing (SIN) task in TRECVID 2011 project aims to identify the correct semantic concept contained within a video shot. The automatic annotation of semantic concepts within video shots can be a fundamental technology for filtering, categorization, browsing, searching, and other video exploitation. New technical issues to be addressed include developing robust methods that adapt to the increasing size and diversity of the video collection.

In TRECVID 2011, there are totally 346 high-level semantic concepts in the semantic indexing task. Comparing with the 130 semantic concepts in last year's task, the number of semantic concepts has greatly increased, posing more challenges to the SIN task. In addition, the total size of video collection also increases more than $1/3$. The participants are allowed to submit a maximum of $2,000$ possible shots for each of the 346 semantic concepts, and the submission result is evaluated using a measure called mean inferred average precision.

In the Instance Search (INS) task, a collection of sample queries are provided that may depict a particular person, object or location in some video clips. Each query image is also attached with a binary mask to indicate the region of interests. The participants are required to find those video segments that contain the person/object/location shown in a given sample query from the testing clips. Up to $1,000$ clips can be submitted for final evaluation in terms of mean inferred average precision.

This paper is organized as follows. Section 2 describes our proposed framework and how it handles the challenges in the TRECVID 2011 SIN task. Section 3 illustrates the way of dealing with TRECVID 2011 INS task. Section 4 concludes the whole paper and proposes some future directions.

2 Semantic Indexing (SIN)

The overall SIN framework (shown in Figure 1) aims to bridge the semantic gap between low-level features and high-level semantic concepts. In TRECVID 2011 SIN task, one key frame per shot is extracted by NIST data provider for both the training and the testing videos. Then, we extract 361 global and local visual features from these key frames, including color, shape, texture, Local Binary Patterns (LBP), etc. SMR is used to retrieve the instances belonging to the target concept. The SMR model generats a baseline result, which is later refined by the other models learned from the meta-data, audio, and object-level information.

2.1 Subspace Modeling and Ranking

Considering the huge amount of data instances in TRECVID 2011 SIN task, a light-version of the SMR method proposed in [2] is used. In the training step of SMR, the z-score normalization (as shown in Equations (1) and (2)) is applied to the positive training instances and the negative training instances, respectively.

$$PosX = \frac{X^{pos} - \mu^{pos}}{\sigma^{pos}}; \tag{1}$$

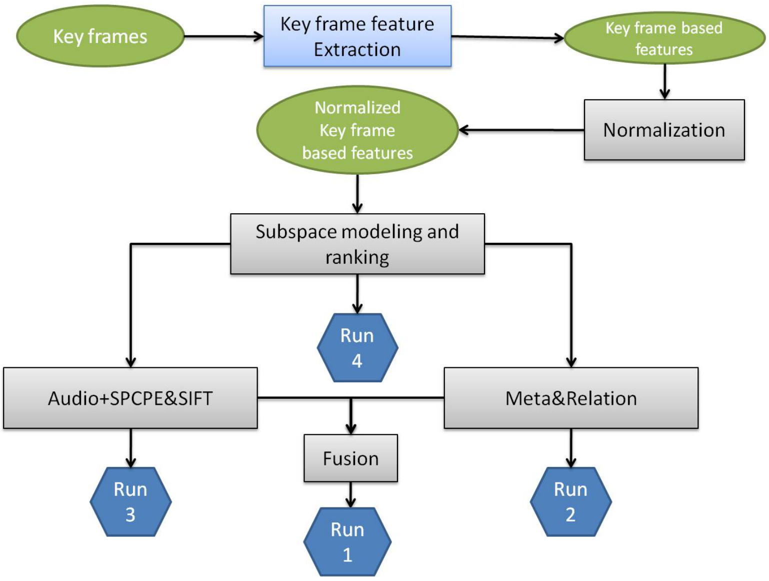

Figure 1. The whole framework for semantic indexing

$$NegX = \frac{X^{neg} - \mu^{neg}}{\sigma^{neg}}, \tag{2}$$

where μ^{pos} and σ^{pos} values are the sample mean value and standard deviation of the positive training instances and μ^{neg} and σ^{neg} values are the sample mean value and standard deviation of the negative training instances. Then, Singular Value Decomposition (SVD) is used to derive the Principal Components (PCs) and eigenvalues of the normalized positive instances (denoted by *PosX*) and those of the normalized negative instances (denoted by *NegX*) from their covariance matrix *CovPosX* (see Equation 3) and *CovNegX* (see Equation 4), respectively.

$$CovPosX = \frac{1}{N_{pos}} PosX^T \cdot PosX, \tag{3}$$

$$CovPosX = \frac{1}{N_{neg}} NegX^T \cdot NegX, \tag{4}$$

where N_{pos} and N_{neg} are the number of positive instances and negative instances, and $PosX^T$ and $PosY^T$ are the transpose of $PosX$ and $PosY$, respectively. Equation (5) shows applying SVD on *CovPosX* with the eigenvalues $\lambda_1^{pos} \geq \lambda_2^{pos} \geq \cdots$.

$$CovPosX = U_{pos} \Sigma_{pos} V_{pos}^*. \tag{5}$$

Here, U_{pos}=$\{PC_1^{pos}, PC_2^{pos}, ...\}$ and the diagonal value of Σ_{pos} is $\{\lambda_1^{pos}, \lambda_2^{pos}, ...\}$. U_{neg} and Σ_{neg} can be derived in the same manner. Those PCs attached to zero eigenvalues are discarded since they contain no extra information. A subspace spanned by U_{pos} is built for the positive training instances and likewise a subspace spanned by U_{neg} is built for the negative training instances. The two subspaces as well as those related eigenvalues are used in the testing phase for each testing instance (to be shown later).

In the testing phase, each testing instance X_{test} goes through the normalization step using the pairs (μ^{pos}, σ^{pos}) and (μ^{neg}, σ^{neg}) (see Equation (1) and Equation (2)) to get $PosX_{test}$ and $NegX_{test}$. Then, $PosX_{test}$ is projected to the subspace spanned by U_{pos}, and $NegX_{test}$ is projected to the one spanned by U_{neg}, as shown in Equation (6) and Equation (7).

$$Y_i^{pos} = PosX_{test} \cdot PC_i^{pos}; i \in [1, \text{\# of PCs in } U_{pos}] \tag{6}$$

$$Y_j^{neg} = NegX_{test} \cdot PC_j^{neg}; j \in [1, \text{\# of PCs in } U_{neg}] \tag{7}$$

The dissimilarity measures shown in Equation (8) and Equation (9) are used to calculate the dissimilarity of the projected data from Equation (6) and Equation (7) to positive and negative models.

$$DisX_{pos} = \sum_i \frac{Y_i^{pos} \cdot Y_i^{pos}}{\lambda_i^{pos}}, i \in [1, \text{\# of PCs in } U_{pos}] \tag{8}$$

$$DisX_{neg} = \sum_j \frac{Y_j^{neg} \cdot Y_j^{neg}}{\lambda_j^{neg}}, j \in [1, \text{\# of PCs in } U_{neg}] \tag{9}$$

The idea behind these similarity measures is that an instance fits to a model if the dissimilarity value calculated from the model is small. Based on this idea, a ranking strategy is proposed in Equation (10).

$$SC = \frac{DisX_{neg} - DisX_{pos}}{DisX_{neg} + DisX_{pos}}. \tag{10}$$

This ranking strategy implies that an instance closer to the positive learning model than to the negative learning model must have a larger possibility to belong to the positive class. Therefore, for a testing instance, the higher it holds a SC value, the closer it is towards the positive model. Therefore, it should get a higher rank.

2.2 Meta and Relation

Compared with the visual features, the text features are considered as high-level features. Thus, a model learned from the text features can suffer less from the semantic gaps. Since each video is associated with some meta data which are stored in the structured xml files, it motivates us to use the fields such as "title", "description", and "subject" as meta information to facilitate content-based semantic detection. In order to extract useful information from "title" and "description", each word in "title" and "description" is considered as a term. Stop word removal is then adopted to remove the stop words such as "the", "is", and all the numbers. A list of 561 manually collected stop words is built and all the words in "title" and "description" that appear in the list are removed. Since the videos are uploaded by people from everywhere, the xml files may contain different languages. Wordnet [3] is then used to validate the terms and only those existed in Wordnet (English words) are kept. Next, those words that are considered very rare and very common are also removed if their frequency counts are below the predefined lower bound or above the predefined upper since these words do not have much discriminative information. Different from the regular processing in text mining, the frequency counts of the terms among videos instead of the term frequency values are used. This is out of the concern that the first appearance of a term in a video is more important, but successive appearances do not contribute much. According to the characteristics of this year's data (such as the video amount and the number of positive videos in the training set), we consider a term as a rare word if the number of training videos containing this term is smaller than 2, and consider a term as a common word if the number of training videos containing this term is more than 500.

After these pre-processing steps, $4,779$ terms are extracted as textual features. Since these meta data are considered as video level information, each video is treated as an instance. If the meta data of a video contain a term, then value 1 is given to this feature for this instance, otherwise 0 is given. To get the video-level training labels, we simply repeat the shot-level labels. It means that if a shot in the video is labeled as positive, then this video is also labeled as positive. If none of the shots in a video is positive, then this video is regarded as negative. After generating the video-level labels, a LibSVM [4] classifier is trained using these textual features and generates the ranking scores for the testing videos.

Figure 2 shows the whole procedure of training a textual feature based model and generating the video-level LibSVM ranking scores. The ranking scores of the testing videos are then fused with the ranking scores from the

KF model to generate the fused ranking scores. Again, we need to repeat the video-level LibSVM scores in order to match with the shot-level KF scores. So the final score of a testing video shot is the summation of its KF score and the LibSVM score of the video it belongs to.

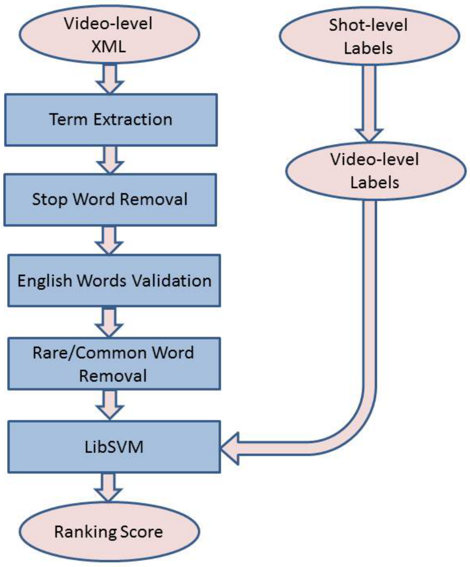

Figure 2. The video-level model using meta data

In addition to the meta data, the relationships between concepts provided by NIST are also taken into account to refine the results of some concepts. Some concepts are usually too general to be easily detected, like "Animal", "Ground Vehicles", and "Outdoor"; while good performance can often be achieved on some other concepts that are relatively specific, such as "Cats", "Pickup_Truck" and "Clouds". Such observations motivate us to use the specific concepts with good results (also called the base concepts) to refine the results of the general concepts with poor results (also called the implied concepts). NIST provides two kinds of pair-wise relations if they exist between two concepts: "implies" and "excludes". For example, "Cityscape" implies "Outdoor" and "Indoor" excludes "Outdoor". Only the "implies" relation is considered in our framework. The question now is how to find out the base concepts and the implied concepts. We use the TRECVID 2010 training data as the training set and the TRECVID 2010 test data as the validation set. The result for each concept on the validation set can then be used to select the base concepts and implied concepts. Table 1 shows the concepts that are refined according to the "implies" relation and the validation results.

2.3 Audio, SPCPE and SIFT

Mel-Frequency Cepstral Coefficients (MFCCs) [5] features are extracted from the audio data within each shot. These audio features approximate the human auditory system's response and have been proved to be effective for speech and music discrimination [6]. Specifically, for each audio frame, 12 MFCC coefficients and C_0 are computed to form a feature vector. In addition, the first-order and second-order derivatives over 10 frames are computed and appended to the original feature vectors. Therefore, the final feature vectors have a dimension of 39. Afterwards, the Gaussian Mixture Models (GMM) are trained using the MFCC feature vectors to annotate music, noise or speech for the audio clips. The audio model specifically targets at some concepts in this task, such as dancing, instrumental musician and singing.

For some concepts that particularly describe objects, such as airplane, animals, and stadium, an unsupervised video segmentation method called the Simultaneous Partition and Class Parameter Estimation (SPCPE) algo-

Table 1. Relations utilized to refine results

Base Concepts	Implied Concepts
Cats, Quadruped	Animals
Canoe	Boat_Ship
Politicians	Politics
Daytime_Outdoor, Beach, Cityscape, Mountain, Sky, Clouds, Lakes, Oceans, River, Valleys	Outdoor
Baseball, Ski	Sports
Car, Truck, Pickup_Truck	Ground_Vehicles
Boy, Girl	Child
Plant, Animal, Tree, Flowers, Forest	Eukaryotic_Organism
Throw_Ball	Throwing
Ski	Snow
Car, Truck, Pickup_Truck	Vehicles

rithm [7] is applied to separate the objects from the image background and then the scale-invariant feature transform (SIFT) [8] is employed to get interesting points in the object regions. We think interesting points in objects region can reflect more precise characteristics of the concept than those in the full image region. The matching of the interesting points towards all the training key frames is performed for each testing instance (testing key frame). The number of matching points are then transferred to the ranking score for each testing key frame.

2.4 Experimental Results

Figure 3 to Figure 6 show the performance of our semantic indexing results. More clearly, Table 2 shows the mean average precision values of the first 10, 100, 1000 and 2000 shots. The inferred true shots and mean inferred average precision are shown in Table 3.

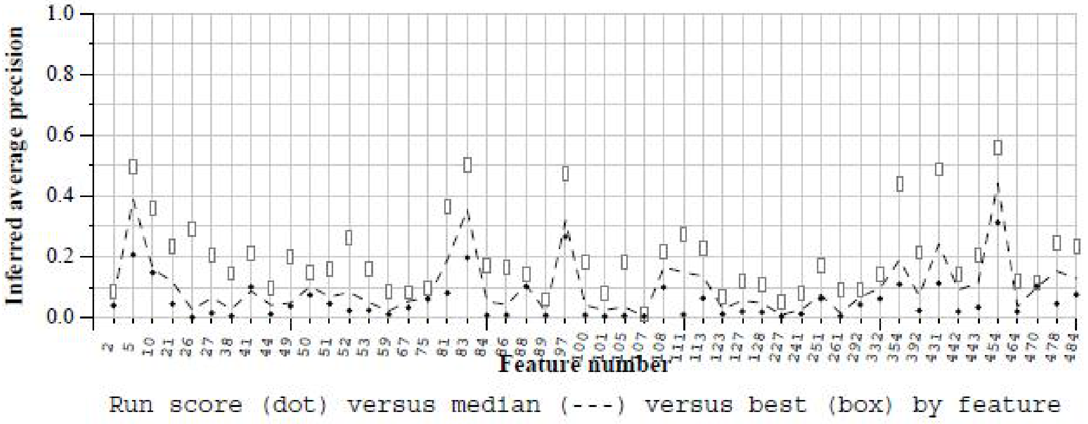

Figure 3. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-1_1*

Evaluation results show that more features (including visual, text, audio, and object-level information) provide only limited helps to improve the retrieval results. It can be observed from these four different runs that their performance results are quite close. This is due to the fact that the fusion strategy is not applied to all concepts. In

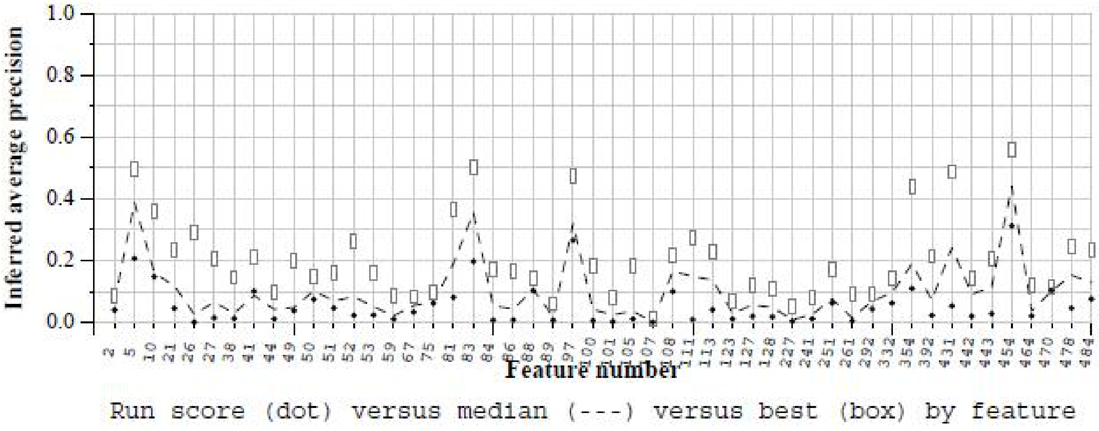

Figure 4. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-2_2*

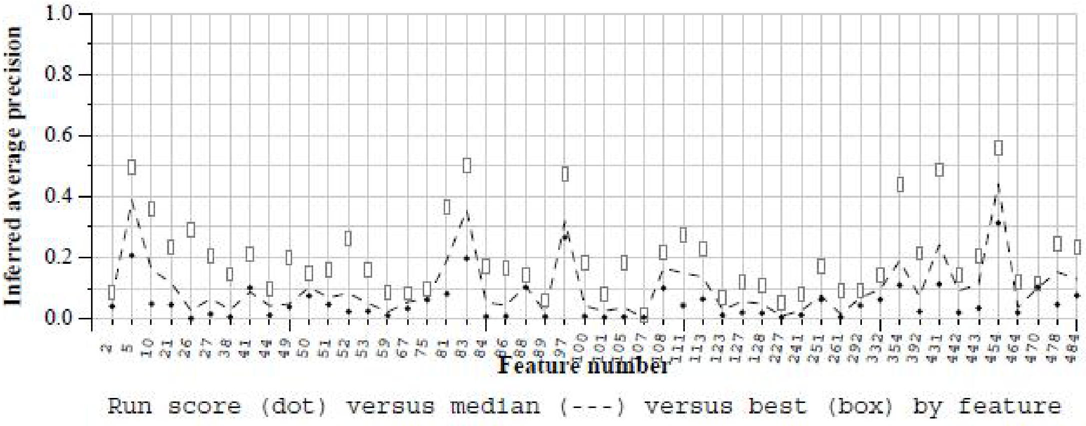

Figure 5. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-3_3*

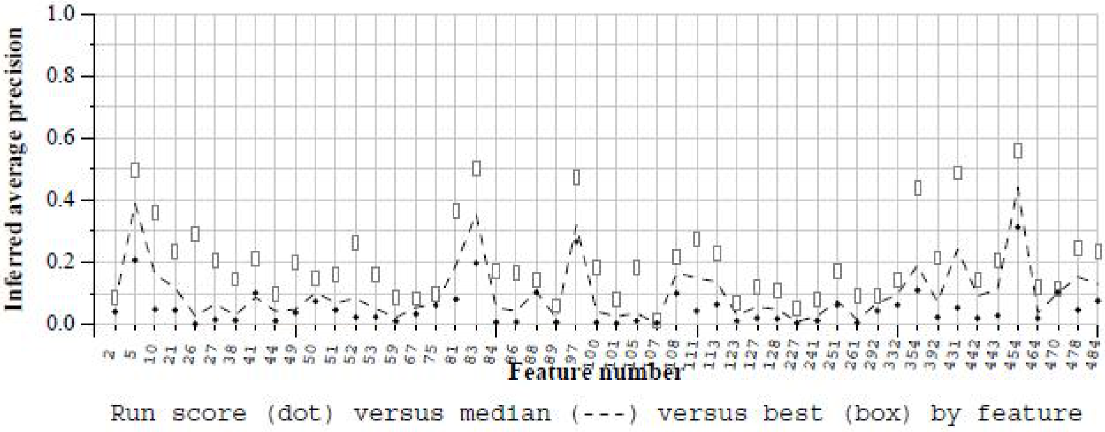

Figure 6. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-4_4*

Table 2. The mean average precision values at first n shots for all four runs

n	10	100	1000	2000
F_A_FIU-UM-1_1	53.9%	42.1%	27.1%	23.8%
F_A_FIU-UM-2_2	51.9%	40.6%	27.0%	23.7%
F_A_FIU-UM-3_3	54.1%	41.8%	27.1%	23.9%
F_A_FIU-UM-4_4	54.1%	40.7%	27.1%	23.9%

Table 3. Inferred true shots and mean inferred average precision

	Inferred true shots	Mean inferred average precision
F_A_FIU-UM-1_1	23794	0.057
F_A_FIU-UM-2_2	23746	0.055
F_A_FIU-UM-3_3	23895	0.056
F_A_FIU-UM-4_4	23855	0.054

fact, we divide the training data from the TRECVID 2011 video collections into two subsets. One subset contains the TRECVID 2010 training data and the other consists of TRECVID 2010 testing data. Evaluations are performed for all 346 concepts to decide whether or not a certain fusion strategy mentioned earlier should be utilized. Since TRECVID SIN task only selects 50 out of 346 concepts for performance evaluation, it is possible that those concepts whose performance is greatly improved were not chosen. However, the current results demonstrate that adding features from different sources can potentially enhance the performance of retrieving some semantic concepts. Therefore, more efforts still need to be made to refine each model within our framework and more fusion methods need to be studied to effectively combine the results from different models.

3 Instance Search (INS)

One of the main different characteristics between the INS task and SIN task is the size of available training data. INS provides two to six query images for each topic, while SIN provides a large training data set for the participants to build the concept models. The subspace approach, SMR, utilized in SIN is not applicable for the INS task, since the limited training data cannot build good subspace models. Therefore, in this task, MCA is employed as the primary approach to train the topic models; while other methods (such as SIFT, KNN, and SIN models) are used to improve the results in the re-ranking step. In addition, we collect 261 extra images together with the provided 95 example images to train the models in order to evaluate the influence of the supplemental data to the searching performance.

Four runs are submitted for this task, which aims at testing: a) the performance of the MCA algorithm in the instance search task; b) the effectiveness of the re-ranking strategy by using the scores from different algorithms; and c) the influence of introducing extra training images to build the topic models. Our framework consists of four modules, namely data pre-processing, feature extraction, classification, and re-ranking. The functionality of each module as well as the estimation of their relative contributions to the results are presented as follows.

3.1 Data Pre-processing

The data pre-processing module is composed of three steps: extra training data collection, object region extraction from the training images, and temporal sampling of the testing clips.

1. Extra Training Data Collection

For the purpose of evaluation and comparison, we collect extra images on certain topics/targets (e.g., setting sun, the Parthenon, and airplane-shaped balloon) from the Internet. For each topic/target, up to 20 extra images are collected based on the description and example images of the topic. This step is to increase the number of positive instances so that a reasonably good model for that topic/target can be trained. As to those difficult topics/targets (e.g., upstairs of windmill, male presenter Y, etc.), no extra images could be found since no enough information can be utilized to find the images with exactly the same topic/target from the Internet.

In summary, for 16 out of 25 topics, 261 extra images are collected to enlarge the training data set. For these collected images, they are adjusted to have the same size and length/width ratio as the example images. Then, image processing tools are used to get the mask of the topics/targets. Therefore, these extra collected images can be regarded as an extension of the provided 95 example images.

2. Object Region Extraction from Training Images

   The masks of the target (e.g., folk, monkey, etc.) are either given by the organizers or generated by ourselves as explained in the previous step. Such information enable the approaches to process the object and background separately. To fully utilize this information, we use the masks and the images to generate the object images and background images for feature extraction. For the testing data whose mask is not provided by the organizer, the SPCPE algorithm is applied to obtain the mask of the object/target in each testing frame.

3. Temporal Sampling of Testing Videos

   For the testing video clips, a multi-frame sampling method is adopted to ensure that the selected frames have enough coverage of the original content. The length of testing video clips is around 2 to 30 seconds and these videos have relatively small content change comparing with those in the SIN task. Here, we averagely sample up to 3 frames to represent the visual content in the clip.

3.2 Feature Extraction

Two categories of features are extracted from the training images and the testing frames, namely texture-based features and SIFT features. The texture-based features are the same as those used in the SIN models, which include information of edge histogram, texture co-occurrence, and texture wavelet. The MCA classifier and the KNN method use the texture features of the images for classification.

On the other hand, the SIFT features have different characteristics which are not suitable for the MCA classifier or the KNN method for testing. Here, the SIFT features used are the version described by Lowe [8], which have n*128-dimension and n is the number of interesting points in the frame. The number of interesting points detected on the object regions varies among different frames/images.

3.3 Classification

Four classification methods are utilized in our four runs. The description of these classifiers are given as follows.

1. MCA Classifier

   Multiple Correspondence Analysis (MCA) is a data analysis technique for nominal categorical data, which is used to detect and represent the underlying structures in a data set. MCA is an extension of the simple Correspondence Analysis (CA) in that it is applicable to a large set of categorical variables [9]. MCA represents data as points in a low-dimensional Euclidean space.

   The functionality of MCA has motivated us to explore its utilization to capture the correspondence between the feature-value pairs and the semantic classes. The similarity of each feature-value pair to a class can be represented by the angle between them in the projected subspace: a smaller angle between a feature-value pair and a semantic class indicates a higher correlation. The detailed description of the MCA classification model can be found in [10].

2. K-nearest neighbor (KNN)

   KNN algorithm is one of the simplest machine learning algorithms: an object is classified by a majority voting of its neighbors, with the object being assigned to the most frequent class amongst its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. Since the training images and a testing clip may come from the same video, we explore the relationships between the object and the background to help the instance search task. Specifically, the features of the object and the background are extracted separately, and then these two feature vectors are concatenated together for the KNN method to generate the classification result.

3. SPCPE+SIFT feature matching

   Since SIFT feature descriptors have achieved good performance in the preceding TRECVID evaluations, in order to explore the ability of the SIFT features on an object region, we employ its descriptors in an image object region instead of the whole image as before. The task of separating objects from background is conducted by an unsupervised video segmentation method called SPCPE, which starts with an arbitrary partition and employs an iterative algorithm to estimate the partition and the class parameters jointly. The SIFT interesting points are then extracted on the separated object region (SPCPE segmentation results) to reduce the number of irrelevant interesting points with regard to the target object. Then, a feature matching step is applied to calculate the matching points for each testing frame. The numbers of matching points are scaled within a range between 0 and 1 to form the SIFT matching scores.

4. Models from semantic indexing task

   We notice that some of the concepts in the semantic indexing task may help the search of a certain topic/target. For example, the concepts female and male can be used to help the person-related search in the instance search task. Therefore, we use these two models to refine those person-related topics in the instance search task by fusing the ranking scores from MCA with the ones from the SIN models.

3.4 Re-ranking

In the INS task, a re-ranking strategy is employed in three runs (i.e., the first, second, and fourth runs) to combine different ranking results from various methods for the better performance; while the third run does not include any re-ranking strategy. Since topic 9024 (upstairs of windmill) and topic 9029 (downstairs of windmill) have lots of local shape information, in the first run, the scaled SIFT matching scores are used to re-rank the MCA results. Those topics that are related to person are re-ranked by the ranking scores from the SIN female or male model which are built from the training data set of the TRECVID 2011 SIN task. The remaining topics use the KNN scores to re-rank.

The second run uses the MCA model built on the 95 training images to re-rank the MCA ranking results from the 356 training images (where 95 of them were provided by NIST and 261 are manually collected by us). This run aims to test the ability of the MCA classifier without the help of other classification approaches.

The fourth run uses the KNN scores from the 95 training images to re-rank the results from the MCA model built from the 95 sample images. This run aims to test the performance trained by 95 example images from multiple classification methods.

4 Conclusion

In this notebook paper, the summary of the work completed by team FIU-UM in TRECVID 2011 semantic indexing and instance search tasks is introduced. The evaluation results show that there are still lots of potential improvements for our current framework. Among them, the following directions will be investigated in the near future.

- Incorporate an optimization step in the SMR model.
- Utilize the object level information to refine the detection results of those concept related to objects.
- Explore the information hidden in the meta data to improve the baseline retrieval results.
- Explore the inter-concept relationships to refine the ranking performance from each individual model.

References

[1] A. F. Smeaton, P. Over, and W. Kraaij, *High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements*, 1st ed. Springer US, 2009.

[2] M.-L. Shu, C. Chen, and S.-C. Chen, “Multi-class classification via subspace modeling,” *International Journal of Semantic Computing*, vol. 5, no. 1, pp. 55–78, 2011.

[3] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[4] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[5] W. Han, C.-F. Chan, C.-S. Choy, and K. P. Pun, “An efficient mfcc extraction method in speech recognition,” in *Proceedings of 2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp. 145–148.

[6] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas, “A comparison of features for speech, music discrimination,” in *Proceedings of 1999 IEEE International Conference on Acoustics, Speech, and Signal*, Mar. 1999, pp. 149–152.

[7] S.-C. Chen, S. Sista, M.-L. Shyu, and R. L. Kashyap, “An indexing and searching structure for multimedia database systems,” in *IS&T/SPIE Conference on Storage and Retrieval for Media Databases*, 2000, pp. 262–270.

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

[9] M. J. Greenacre and J. Blasius, *Multiple Correspondence Analysis and Related Methods*, 1st ed. Chapman and Hall/CRC, 2006.

[10] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, “Correlation-based video semantic concept detection using multiple correspondence analysis,” in *IEEE International Symposium on Multimedia (ISM08)*, Dec. 2008, pp. 316–321.