Florida International University and University of Miami TRECVID 2010 - Semantic Indexing

Chao Chen, Qiusha Zhu, Dianting Liu, Tao Meng, Lin Lin, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33146, USA
{c.chen15, q.zhu2, d.liu4, t.meng, l.lin2}@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

Abstract

This paper presents the framework and results of team Florida International University - University of Miami (FIU-UM) for the semantic indexing task of TRECVID 2010. In this task, we submitted four runs of results:

- F_A_FIU-UM-1_1*: KF+RERANK - apply subspace learning and classification on the key frame-based low-level features (KF) and use co-occurrence probability re-ranking method (RERANK) to generate the final ranked results.*
- F_A_FIU-UM-2_2*: LF+KF+SF+RERANK - apply subspace learning and classification on the key frame-based low-level features (KF) and shot-based low-level features (SF) separately. Then co-occurrence probability re-ranking method (RERANK) is used for both key frame based model and shot based model. Finally, a Late Fusion (LF) step combines ranking scores from each model and generates the final ranked shots.*
- F_A_FIU-UM-3_3*: EF+KF+SF+RERANK - apply subspace learning and classification on combined features from the selected key frame-based low-level features (KF) and shot based low-level features (SF) in the Early Fusion (EF) step. Then co-occurrence probability re-ranking method (RERANK) is used.*
- F_A_FIU-UM-4_4*: SF+RERANK - learning and classification based on shot based low-level features (SF). Then co-occurrence probability re-ranking method (RERANK) is used.*

From the results of different runs, it can be observed that F_A_FIU-UM-1_1 *and* F_A_FIU-UM-3_3 *have better performance than* F_A_FIU-UM-2_2 *and* F_A_FIU-UM-4_4*. It implies that adding features* *from different sources could enhance the effectiveness of the learning and classification model and also visual features seem to be more reliable than audio features for most semantics in TRECVID 2010.*

The framework aims to handle several challenges in semantic indexing. For the challenge of data imbalance, Multiple Correspondence Analysis (MCA) based pruning method is able to reduce the high ratio between the number of negative instances and the number of positive instances. Meanwhile, for the challenge of semantic gap, the proposed subspace learning and ranking method has adopted a new way to select Principal Components (PCs), which spans a subspace where all instances are projected and classification rules are generated. The scores from one-class positive and negative learning models are further used to rank the classified instances. Then the co-occurrence probability re-ranking approach is utilized to improve the relevance of the retrieved shots. Please note that there is one run that adopts late fusion to combine the scores from key frame-based model and shot-based model. Evaluation results show that more efforts still need to be done to refine each module within our framework and some future directions to be explored are discussed in the conclusion section.

1 Introduction

The semantic indexing task in the TRECVID 2010 project can be regarded as an approach of content-based multimedia retrieval. For each high-level semantic, a maximum of 2000 retrieved shots are allowed to be submitted. With regards to content-based multimedia retrieval, there are a few challenges that need to be carefully addressed, and the way to handle these challenges is closely related to the effectiveness of the content-based multimedia retrieval approaches, which further impacts the relevance of the retrieved shots.

The first and fundamental challenge is feature extraction. Top teams like MediaMill and Columbia both have spent a lot of efforts in seeking low-level discriminative features to represent images in video collections. Prior to feature extraction, MediaMill even performs a few sampling processes, such as temporal multi-frame selection to increase the possibility of detecting those semantics that might not be in a single key frame within one shot [1]. Usually, the extracted features are low-level features, including visual features, audio features, key words, etc. The dimensionality of the extracted features is usually huge and may require clustering to build a word vocabulary or codebook library to represent the characteristics of the video data. Due to the huge amount of multimedia data in the TRECVID project, the efficiency of the feature extraction algorithms as well as the computational power of the machines become critical issues. For example, MediaMill in TRECVID 2009 reported to have processed a total of one million I-frames utilizing a GPU implementation of visual feature extraction methods.

Data imbalance is another issue in multimedia research. The ratio of the number of negative samples to the number of positive samples in the TRECVID 2009 and 2010 projects on average is more than 100:1. Therefore, many research teams participating in the TRECVID 2009 project, such as the team of IBM and Fudan University, the team of THU and ICRC, and the team of Peking University and Intel, proposed approaches to address the data imbalance issue. The prevailing methods would be the under-sampling and over-sampling techniques. For example, team Peking University [2], one of the teams that achieved top results for high-level semantic extraction task, has adopted OnUm method that combines over-sampled positive samples and under-sampled negative samples together to form a group of subsets. Within these subsets, the ratio is quite balanced and the final scores are generated from these subsets. On the other hand, team MediaMill fixes the weights of the positive and negative data according to their distributions in the training set to compensate the imbalanced ratio between the number of positive instances and the number of negative instances.

Among all the challenges, “semantic gap” can be considered the most attractive issue. Many approaches have been developed to bridge the gap between the extracted low-level features and the targeted high-level semantics. The existing approaches usually rely on relevance feedback and/or learning algorithms. By applying relevance feedback, the retrieved results can be improved iteratively. However, it is sometimes not easy, if not impossible, to get users’ feedback. Therefore, learning algorithms have gained increasing attentions, especially in the situation where the feedback method is hardly applicable. Lots of learning algorithms have been developed in the literature. In the multimedia retrieval domain, the prevailing algorithms are Support Vector Machines (SVM) and their variations. In the previous TRECVID projects, most of the teams with top ranking results all utilized Support Vector Machines as their learning algorithms. The kernel of SVM could be radial basis function or chi-square. The former kernel function usually performs better than the other kernels; while the latter kernel function gains increasing attentions after the chi-square kernel function is utilized in [3].

In addition, ranking and re-ranking strategies also play an important role in the semantic indexing task. Ranking and re-ranking aim to provide the end users the most relevant retrieved results as many as possible. Ranking could be broadly categorized into pairwise ranking [4] and model-based ranking [5]. Since the number of generated instances (or samples) with audiovisual features representing each shot in the TRECVID videos is too large, it is very impractical to apply pairwise ranking algorithm in the TRECVID project. Instead, the model-based ranking method is extensively used in the semantic indexing task. These ranking methods are usually integrated in the learning and classification step that originally aims to handle the semantic gap problem. Re-ranking tries to further refine the ranked results by considering some other relevant auxiliary information. For example, information such as the ranking score from a different model (such as face detection) can help further accurately index the target semantic.

In this paper, each module of our semantic indexing framework is introduced and the solution to each of the aforementioned issues is discussed one by one. The paper is organized as follows. Section 2 describes our proposed framework and how it handles the challenges in the TRECVID 2010 semantic indexing task. Section 3 shows our experimental results and Section 4 concludes this paper and points out some future directions.

2 The Proposed Semantic Indexing Framework

The proposed semantic indexing framework is shown in Figure 1. Each run has been marked by a number in the framework. Run 1 uses key frame-based features exclusively; while Run 4 uses only shot-based features. Run 3 combines these two categories of features together and finally Run 2 is the fusion of Run 1 and Run 4. The proposed framework contains a set of modules. Each module aims to handle particular problems that were discussed in the previous section.

2.1 Feature extraction

Two different categories of features have been extracted from the TRECVID 2010 video collections, namely key frame-based features and shot-based features. In the set of key frame-based features, the following features are extracted: color dominant in the RGB color space, color histogram in the HSV space,

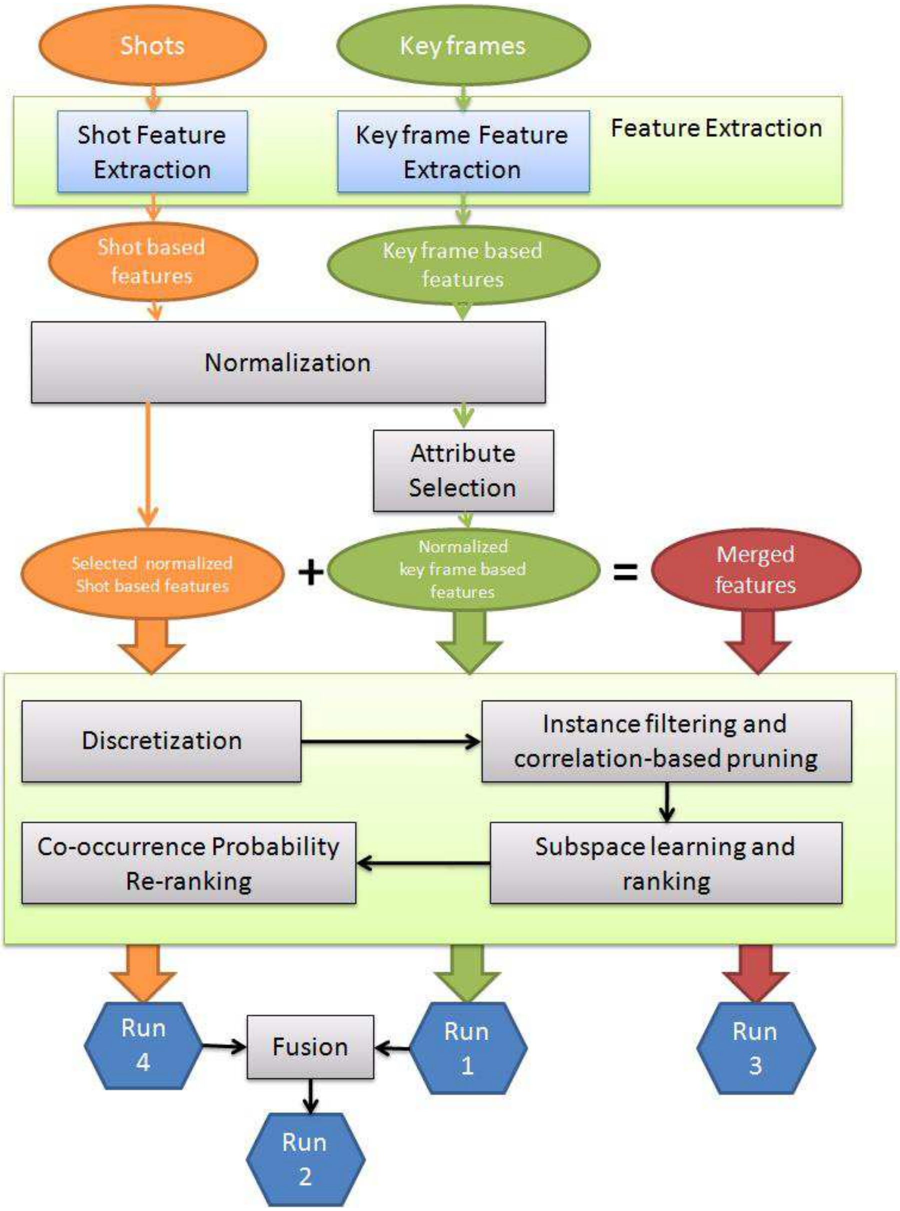
Shots
Key frames
Shot Feature Extraction
Key frame Feature Extraction
Feature Extraction
Shot based features
Key frame based features
Normalization
Attribute Selection
Selected normalized Shot based features
+
Normalized key frame based features
=
Merged features
Discretization
Instance filtering and correlation-based pruning
Co-occurrence Probability Re-ranking
Subspace learning and ranking
Run 4
Fusion
Run 1
Run 3
Run 2

Figure 1. The Proposed Framework.

color moment in the YCbCr space, edge histogram, texture co-occurrence, texture wavelet, Tamura texture, Gabor texture, and local binary patterns. In addition, some mid-level features from face detection are captured such as the number of faces. A total of 504-dimension attributes are extracted for the key frame feature set.

In the set of shot-based features, audio features (16-dimension) and visual features (4-dimension) are extracted. The audio feature set is composed of volume-related, energy-related, and spectrum-flux-related features as well as the average zero crossing rates. As for visual features, the grass ratio is calculated as a mid-level feature, which is useful for detecting sports-related semantics such as soccer players. Furthermore, a set of motion intensity estimation features are extracted such as the center to corner pixel change ratio.

Both of the two categories of features target at representing/summarizing the videos provided by TRECVID 2010. Video representation is a difficult task in multimedia research. The key frame-based features are commonly used to describe the video content. Global and local key frame-based features are extensively applied and reported to be effective in the multimedia retrieval task. However, key frame-based features cannot capture audio information and some shot-level informations may be missing. Therefore, shot-based features can serve as an auxiliary information source.

2.2 Data preprocessing

There are several subcomponents in the data preprocessing module.

Normalization

Most of the extracted features have different scales. To prevent the small-scale attributes from being dominated by large-scale attributes, a common way is to apply data normalization. There are many normalization functions that could be utilized to scale the data within a common range. For example, the min-max normalization function can ensure the normalized data in the range of $[0, 1]$. Equation (1) to Equation (3) show a list of normalization functions [6]. Let X_{min} be the minimum value of X, X_{max} be the maximum value of X, μ be the estimated mean of X, and σ be the estimated standard deviation of X.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}; \tag{1}$$

$$X' = \frac{X - \mu}{\sigma}; \tag{2}$$

$$X' = 0.5[tanh\frac{0.01 \cdot (X - \mu)}{\sigma} + 1]. \tag{3}$$

All of these three normalization methods had been applied to the extracted features. After we further did cross-validation on these three different normalized data, we observed that Z-score normalization in Equation (2) rendered the best learning results.

Attribute selection

Attribute selection or sometimes called feature selection is applied only to the key frame-based features since the dimensionality of the key frame-based features is quite large. The input data has more than 500 key frame-based features and there could be a lot of redundant or irrelevant features within the extracted key frame-based features to different high-level semantics. Therefore, attribute selection provides a way to find a set of relevant attributes. The direct benefit from the attribute selection method is that the model can be induced faster since the dimensionality of the attributes has been reduced. In our work, 20 most relevant attributes are selected from the whole 504-dimension key frame-based features and it can be observed that the accuracy and training time both benefit from this technique. However, one thing that needs to be mentioned is that there is a trade-off between the number of attributes retained and the performance of the learning model. Indeed, attribute selection could enhance the learning model by removing irrelevant or redundant attributes. However, if too many attributes have been removed, the performance of the learning model will become poor or over-fitting to the training data. It is one of our future directions to explore the optimal trade-off point within the attribute selection step.

Discretization

Discretization maps the continuous data into the discretized form and could facilitate the learning process [7]. It could reduce the values of features and simplify the representation of the data. Meanwhile, it can also be used as the preprocessing step of the learning methods which do not accept continuous/numeric inputs. In general, discretization methods fall into two categories: one is unsupervised methods such as equal-width and equal-frequency discretization, and the other one is supervised methods such as entropy-based Information Entropy Maximization (IEM) and statistics-based Chi-square. The method used in our framework is a modified version based on the Minimum Description Length (MDL) [8]. Usually, it is an NP-hard problem to find the optimal discretization and sometimes the generated intervals can range from one to more than 100. Therefore, it is not suitable to directly apply original MDL methods. The IEM method we used in TRECVID 2010 has been modified to discretize a feature into a range of intervals between 2 and 10. That is, two intervals will be forced to generated if only one interval is generated by the original MDL method, and at the same time, only 10 intervals will be generated if the original MDL method creates more than 10 intervals. Based on our experiments, the modified version of MDL can reach the same effect as the original MDL while providing better classification results for the learning algorithms like Decision Tree, Ripper, Naive Bayes, AdaBoost, K-Nearest Neighbour, etc.

Instance filtering and correlation-based pruning

In the TRECVID semantic indexing task, a notable characteristic of the data is that the number of negative instances is much more than the number of positive ones. Therefore, filtering/pruning is necessary to address this data imbalance issue. The way we handle the data imbalance issue consists of two steps. First, an instance filtering method is applied to reduce the number of the negative instances. Then a correlation-based pruning methods further prunes the negative instances. However, sometimes a few positive instances are also pruned in this step. The reason is that some of the positive instances and negative instances are very much alike, leading to confuse the learning model. Therefore, it is better to not include those positive instances in training the learning model.

In essence, our proposed correlation-based pruning is a method based on Multiple Correspondence Analysis (MCA). MCA is an extension of the standard Correspondence Analysis (CA) to consider more than two variables [9]. Meanwhile, it also appears to be the counterpart of Principal Component Analysis(PCA) for nominal (categorical) data. The functionality of MCA has motivated us to explore its utilization to capture the correspondence between the feature-value pairs and the semantic classes, and further to utilize this correlation to prune the training data set. The similarity of each feature-value pair to a class can be represented by the angle between them in the projected new subspace: a smaller angle between a feature-value pair and a semantic class indicates a higher correlation [10].

From the calculation of the inner product of a feature-value pair (A_{ji}) and a semantic class (C: C_p and C_n stand for positive and negative class respectively); two angles called $angle_{jip}$ and $angle_{jin}$ (from 0 to 180 degrees) of A_{ji} with respect to C_p and C_n could be captured, where i is between 1 and the total number of features I, j varies for each feature from 1 to the number of intervals for each feature, C_p is the target semantic class, and C_n is the non-target semantic class (shown in Equations (4) and (5)).

$$angle_{jip} = arccos(\frac{A_{ji} \cdot C_p}{|A_{ji}| \cdot |C_p|}) \tag{4}$$

$$angle_{jin} = arccos(\frac{A_{ji} \cdot C_n}{|A_{ji}| \cdot |C_n|}) \tag{5}$$

The sum of $angle_{jip}$ and $angle_{jin}$ is 180 degrees. If $angle_{jip}$, the angle between feature-value pair A_{ji} and class label C_p, is smaller than 90 degrees, then A_{ji} has a higher correlation relationship with C_p. On the other hand, if $angle_{jin}$ is smaller than 90 degrees, then A_{ji} has a higher correlation relationship with C_n. If $angle_{jip}$ or $angle_{jin}$ is equal to 90 degrees, it means that A_{ji} has equal correlation relationships with C_p and C_n. Here, one score defined as $score_k$ for each data instance is calculated by the MCA-based correlation, where k is from 1 to the total number of data instances. It is calculated by the sum of its feature-value pair weights $weight_{jip}$ or $weight_{jin}$ (shown in Equation (6)), where the weights are converted from the angle values. In addition, if $angle_{jip}$ is less than a certain threshold $angle_{thres}$, then the positive sign is assigned; while if $angle_{jin}$ is less than the same threshold $angle_{thres}$, then the negative sign is assigned. Otherwise, $angle_{jip}$ or $weight_{jin}$ is set to be 0.

$$score_k = \sum_{i=1}^{l} weight_{jix}, \tag{6}$$

where

- $weight_{jix}$ = $weight_{jip}$ if $angle_{jip} \leq angle_{thres}$, then $weight_{jip}$=$\frac{180-angle_{jip}}{90}$;
- $weight_{jix}$ = $weight_{jin}$ if $angle_{jin} \leq angle_{thres}$, then $weight_{jin}$=$-\frac{180-angle_{jin}}{90}$.

MCA prunes both the training and testing instances [11]. For the training data, when the value of an instance's score is larger than or equal to a threshold, the instance is considered to be positive. Otherwise, it is considered to be negative. If this estimation is in accordance with its class label, then this instance is considered to be a typical/pure training instance; otherwise, it is considered to be a fuzzy/suspected one. Only typical/pure training instances are retained to train the classifiers. For testing instances, on the contrary, those that can be easily classified as positive or negative instances will be filtered out and only fuzzy/suspected instances are kept to get further classified. Thus applying correlation-based pruning can not only increase the classification accuracy but also save the classification time.

It is observed that the pruning of a few positive instances will enhance the F1-score of the learning model. This improvement might be caused by the removal of those confusing positive instances at the boundary where they are mixed with the negative instances. However, one problem occurs during instance filtering, that too many negative instances could be filtered out, leading the learning model become too over-fitting to the training data. Therefore, one of the the future directions is to explore the near-optimal trade-off point within this instance filtering step.

2.3 Subspace learning and ranking

To handle the semantic gap issue in video retrieval, a common way is to employ the learning methods. Most of the participant groups adopt Support Vector Machine (SVM) to build the learning model and further use the model to retrieve a ranking score for each instance. SVM generates hyper-planes that are capable of separating the data instances belonging to different classes in a multidimensional space. The kernel methods allow SVM to handle non-linear cases. The most popular kernel that SVM used in TRECVID is Radial Basis Function, but some top groups have also tried the chi-square kernel. While not applying SVM, our learning method is developed with the aim of building the models for positive and negative instances based on their overall characteristics, including their mean and standard deviation, principal components, etc. In this way, our focus can be shifted to capture the positive and negative instances' behavior globally. The proposed methods project instances into the principal component (PC) subspace. Therefore, it is called subspace modeling in short. In the learning step, two one-class learning models, one for positive instances and the other for negative instances will be built respectively. Later, a ranking step will utilize the scores from both positive and negative models to rank the instances.

The subspace modeling technique requires three steps: normalization, PC projection and classification rule generation. The normalization step is slightly different from the one mentioned previously. The basic statistical information like mean and standard deviation is first derived from the positive and negative data, respectively. Then Z-score normalization will be applied on the training data that is composed of positive and negative instances. However, for the positive model, the μ and σ values are the sample mean value and standard deviation of positive instances while the corresponding parameters in the negative model are the sample mean and standard deviation value of the negative instances. The PC selection methods are improved compared with our previous work [12][13] by selecting several representative PCs according to their separability.

Assume that there is a positive learning model which uses μ and σ estimated from the positive instances to normalize all the training data. It is easy to see that the positive instances after the PC projection hold an interesting property. That is, these positive instances hold a zero mean and their variance is just the same as the eigenvalue corresponding to that PC (see Equation (8)). Assume that the inner class difference could be represented by its variance, we introduce Equation (9) to measure the separation ability of each PC according to Fisher's linear discriminative rule to maximize the inter-class difference while minimizing the inner-class difference.

$$PosX = U\Sigma V^{*}, \tag{7}$$

where U=$\{PC_1, PC_2, ...\}$ and the diagonal value of Σ is $\{\lambda_1, \lambda_2, ...\}$.

$$var[PosY_j] = var[PosX * PC_j] = \lambda_j, \tag{8}$$

where PosX is a positive instance of the training data, PC_j is the j-th PC, and λ_j is the eigenvalue of PC_j. Both PC_j and λ_j are from singular value decomposition of PosX as shown in Equation (7).

$$SEP_j = \frac{H^2(PosY_j, NegY_j)}{\lambda_j}, \tag{9}$$

where H stands for Hellinger distance, $PosY_j$ is the projection of a positive instance of the training data on PC_j, $NegY_j$ is the projection of a negative instance of the training data on PC_j, λ_j is the eigenvalue of PC_j. The Hellinger distance is used to calculate the inter-class difference here and inner-class difference is selected to be the variances that equal to the eigenvalues.

Then all PCs are sorted according to their SEP_j values and the k largest ones are selected as the representative PCs. With these representative PCs, a chi-square distance-based score Dis is calculated using Equation (10).

$$Dis = \sum_j \frac{Y_j^2}{\lambda_j}, \tag{10}$$

where Y_j stands for the projection on PC_j of the whole training data using Equation (11).

$$Y_j = X \cdot PC_j, \tag{11}$$

where X=$\{X_1, X_2, ...\}$ is a combination of the positive instance denoted by PosX and the negative instance denoted by NegX.

Please note that Dis_{pos} is calculated from the positive model as an example here. Correspondingly, there is a Dis_{neg} value for the negative learning model. Based on the two chi-square distance-based scores, the classification rules could be generated. The classification rules are as follows.

- If $Dis_{pos} \leq \beta \cdot Dis_{neg}$, assign positive label to the instance;
- If $Dis_{pos} \geq \beta \cdot Dis_{neg}$, assign negative label to the instance.

Here, β is a weight value that is used to prevent too many false positives in the classification result. The optimal value β can be derived during the model training phase.

The idea behind the rules is that if an instance is closer to the positive learning model than the negative learning model, then it is reasonable to assign the positive label to that instance rather than the negative label. The ranking strategy is based the ranking score defined in Equation (12).

$$SC = Dis_{neg} - Dis_{pos}. \tag{12}$$

For an instance, the higher it holds a SC value, the closer it is towards the positive learning model. Therefore, it should get a higher rank.

2.4 Co-occurrence probability re-ranking and late fusion

Re-ranking methods could help increase the ranking result by taking into more information from the other sources. In TRECVID 2010, two approaches are adopted to further enhance the relevance of the retrieval results. The first method considers the co-occurrence between semantic pairs and the other one combines the ranking scores from different runs.

The first method is called co-occurrence probability re-ranking. It originates from the fact that the semantics could have some relationships with each other. For example, semantic “Vehicle” is closely related to “Road”; while “Swimming” must be accompanied with a “Person”. Thus, a co-occurrence probability matrix CP=CP_{st} is first constructed from the training labels where CP_{st} is defined in Equation (13).

$$CP_{st} = P(C_s|C_t) = \frac{N(C_s, C_t)}{N(C_s)}. \tag{13}$$

Here, $N(C_s)$ means the number of training instances belonging to semantic C_s, and $N(C_s, C_t)$ means the number of training instances belonging to both C_s and C_t. The re-ranking rule is shown in Equation (14).

$$FS_{sk} = \sum_{t=1}^{N} P_{st} \cdot L_{tk}, \tag{14}$$

where FS_{sk} is the final score of the k-th instance of semantic s, and L_{tk}=1 if the k-th instance of semantic t is positive. Otherwise, L_{tk}=0.

The second approach is a late fusion of different runs. Late fusion is based on the scores from shot-based and key frame-based models. First, the score of each model is obtained by subspace-based modeling. Then the two scores are fused using a weighting scheme as given in Equation (15).

$$score_{fusion} = score_{shot} * weight_{shot} + score_{keyframe} * weight_{keyframe}, \tag{15}$$

where the weights, i.e., $weight_{shot}$ and $weight_{keyframe}$, are determined by the $F1$ performance of shot-based and key frame-based models in the training domain. Specifically, the weights are calculated by Equation (16) and Equation (17).

$$weight_{shot} = \frac{F1_{shot}}{F1_{shot} + F1_{keyframe}}; \tag{16}$$

$$weight_{keyframe} = \frac{F1_{keyframe}}{F1_{shot} + F1_{keyframe}}. \tag{17}$$

3 Experiments and Results

The proposed semantic indexing framework utilizes the data provided by TRECVID 2010. No other sources are included in the training pool. The testing data does not need instance filtering but it goes through the correlation-based pruning step where some testing instances are pruned due to their closeness to the negative training instances. The retrieval result of all 130 high-level semantics are submitted and the evaluation results are shown in Figure 2 to Figure 5 and in Table 1. From these four figures and one table, it can be seen that Run 3 gives the best retrieval result followed by Run 1 and Run 2. Run 4 gives the worst result. Compared with other runs, Run 3 combines shot-based low-level features and key frame-based low-level features right after the feature extraction step. This run uses more low-level features than the other 3 runs. It combines 20 key-frame based low-level features with all shot-based low-level features. Run 3 considers the fusion of shot-based information and key-frame based information at the feature level. In contrast, run 2 does the fusion at the model level. It combines the scores from both shot-based learning model and key frame-based learning model. In addition, the observation that run 1 performs much better than run 2 also gives us an impression that key frame-based low-level visual features are more reliable than the shot-based low-level audio features in representing the semantics within videos. The reason behind it might be that the related 130 semantics are more closely related to visual than audio information of the provided videos.

More clearly, Table 1 shows the average precision values of the first 10, 100, 1000 and 2000 shots.

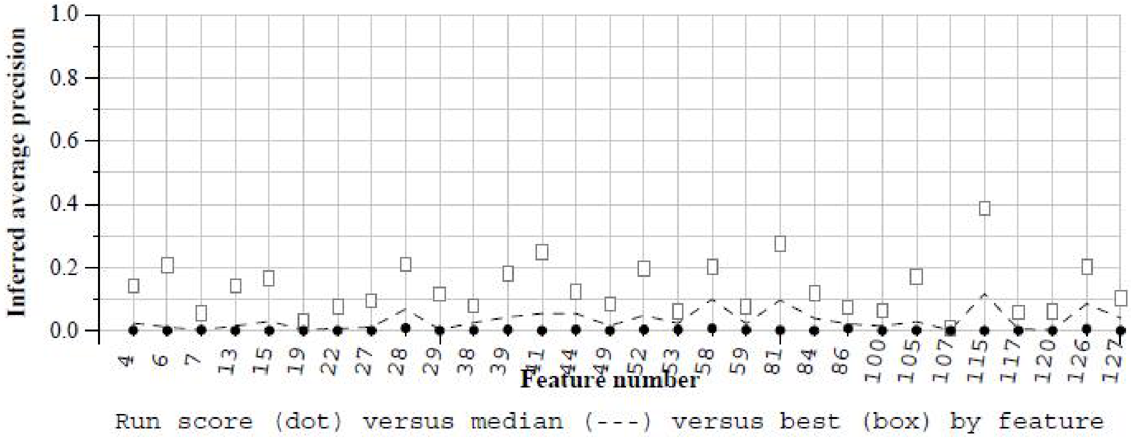

Figure 2. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-1_1*

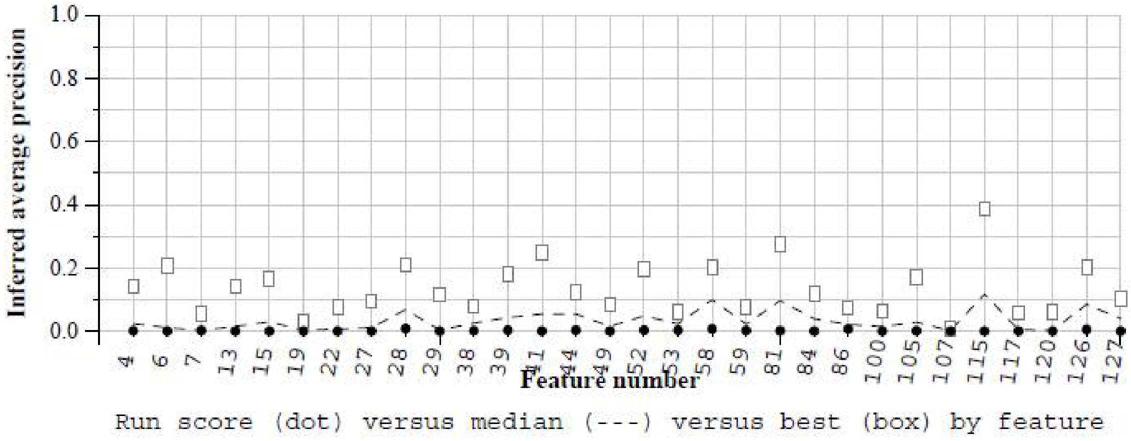

Figure 3. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-2_2*

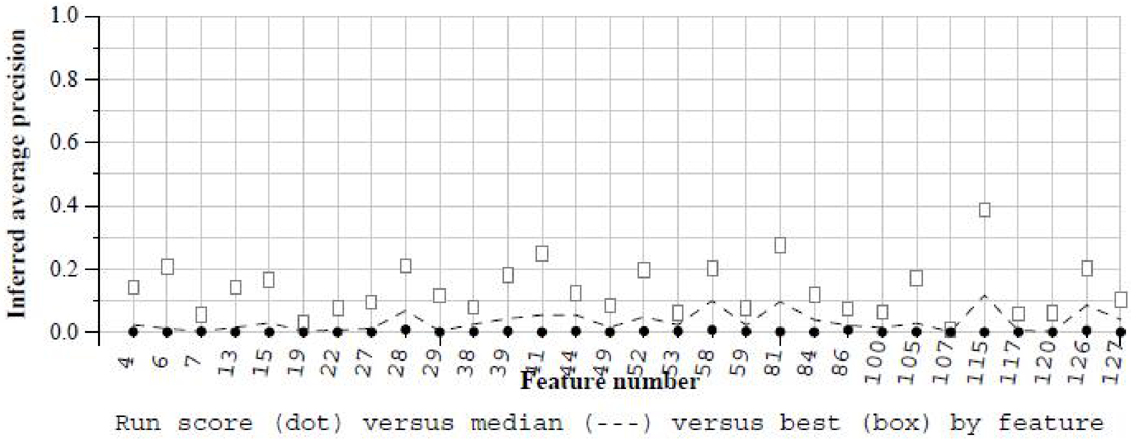

Figure 4. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-2_2*

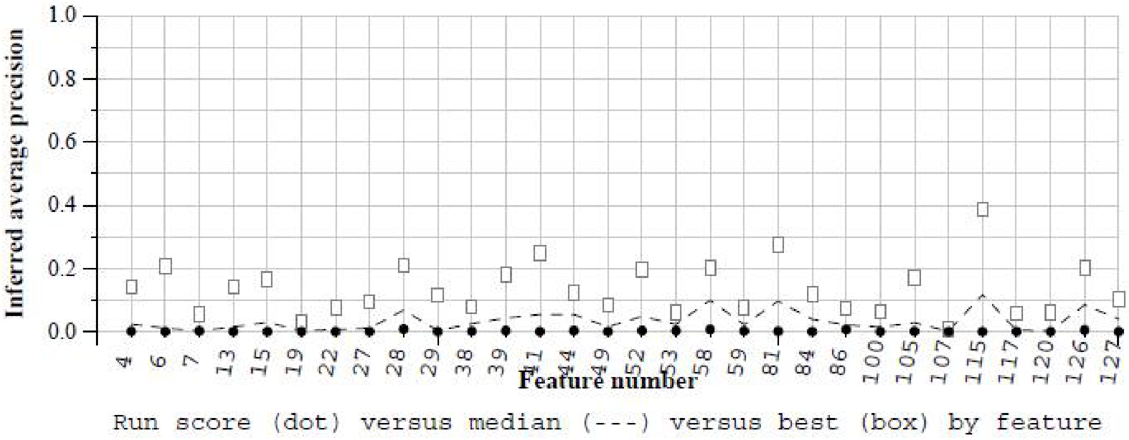

Figure 5. Run scores (dot) versus median (—) versus best (box) for *F_A_FIU-UM-2_2*

Table 1. The average precision values at n shots for four runs

n	10	100	1000	2000
F_A_FIU-UM-1_1	3.3%	4.3%	3.5%	2.7%
F_A_FIU-UM-2_2	3.0%	2.9%	1.9%	1.5%
F_A_FIU-UM-3_3	5.0%	6.3%	3.7%	2.8%
F_A_FIU-UM-4_4	2.0%	2.1%	2.0%	2.1%

4 Conclusion

In the TRECVID 2010 semantic indexing task, a semantic indexing framework is introduced to provide a list of ranked submissions for all 130 high-level semantics. Within our framework, the details of feature extraction, attribute normalization, attribute selection, methods of instance filtering and correlation-based pruning, algorithms of subspace learning and ranking, as well as co-occurrence probability re-ranking approach and late fusion methods are introduced. Our efforts have been spent on the issues like data imbalance, semantic gaps, and ranking strategy. MCA-based correlation pruning reduces the high negative-to-positive ratio of the training instances and helps identify pure instances in the testing set as well. It shows the importance for the later semantic learning and ranking process. Subspace modeling and ranking method provide a way to bridge the semantic gap between low-level features and high-level semantics. Meanwhile, the ranking score provided by subspace modeling is further used for re-ranking and a later fusion method is used to generate a new run.

There are some issues that require further exploration. The first issue is the sufficiency of the current low-level features in representing the video content to capture the characteristics of the video content. The second issue is how many attributes to be retained for each semantic. A trade-off should be made between the improvement of the learning performance and the risk of over-fitting. The third issue is how the negative instances are handled. More negative instances are removed in the instance filtering process. These abandoned negative instances may also contain very important information, without which the learning model will lose some part of its extensibility to the testing instances. The last but not the least issue is the exploration of the learning and classification algorithms. The current subspace modeling algorithm is a linear approach, which has a limitation in handling TRECVID multimedia data. Therefore, the way to handle non-linear case is also a future direction to be explored for our semantic indexing framework.

References

[1] C. Snoek and et al., "The mediamill trecvid 2009 semantic video search engine," in *TRECVID workshop notebook paper*, November 2009.

[2] Y.-X. Peng, Z.-G. Zhang, and et al., "Pku-icst at trecvid 2009: High level feature extraction and search," in *TRECVID workshop notebook paper*, November 2009.

[3] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classifcation of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[4] Y. Liu, T. Mei, J. Tang, X. Wu, and X.-S. Hua, "Graph-based pairwise learning to rank for video search," in *Proceedings of the 15th International Multimedia Modeling Conference on Advances in Multimedia Modeling*, January 2009, pp. 175–184.

[5] C. Steven and R. Michael, "A multimodal and multilevel ranking scheme for large-scale video retrieval," *IEEE Transactions on Multimedia*, vol. 10, no. 4, pp. 607–619, June 2008.

[6] M. Marsico and D. Riccio, "A new data normalization function for multibiometric contexts: A case study," in *Proceedings of the 5th international conference on Image Analysis and Recognition*, June 2008, pp. 1033–1040.

[7] Y. Ying, G. I. Webb, and X.-D. Wu, "Discretization methods," in *Data Mining and Knowledge Discovery Handbook*. Springer US, 2010, pp. 113–130.

[8] P. D. Grunwald, *The Minimum Description Length Principle*, 1st ed. The MIT Press, June 2007.

[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)*, December 2008, pp. 316–321.

[11] L. Lin, C. Chen, and et al., "Florida international university and university of miami trecvid 2009 - high level feature extraction," in *TRECVID workshop notebook paper*, November 2009.

[12] T. Quirino and et al., "Collateral representative subspace projection modeling for supervised classification," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, November 2006, pp. 98–105.

[13] C. Chen, M.-L. Shyu, and S.-C. Chen, "Supervised multi-class classification with adaptive and automatic parameter tuning," in *IEEE International Conference on Information Reuse and Integration (IRI09)*, August 2009, pp. 433–434.