Fudan University at TRECVID 2003

Lide Wu, Yuefei Guo, Xipeng Qiu, Zhe Feng,Jiawei Rong, Wanjun Jin, Danlan Zhou, Rongrong Wang, Ming Jin
Media Computing & Web Intelligence Group, Dept. of Computer Science and Engineering
Fudan University, Shanghai 200433, P. R. China

Abstract

This year we have participated in all the four tasks including shot boundary determination, story segmentation, high-level feature extraction and video search task. Several low-level visual, audio and text features are extracted for each of the tasks. In order to reduce the semantic gap between low-level features and high-level concepts, we have tried to use some learning algorithms, such as Adaboost and Maximum Entropy model, to automatically select and fuse low-level features. In this paper, we present the approaches for each task and discuss some of the evaluation results.

1. Introduction

This year, we participated in four tasks: (1) shot boundary determination, (2) story segmentation, (3) high-level feature extraction and (4) search task.

In Shot Boundary Determination Task, we submitted 6 runs using our old shot segmentation system, which is tuned on this year's data. In Story Segmentation and Classification task, we submitted 10 runs for type A, B and C task. In the high-level feature extraction, we divided 17 high-level features to four categories: scene, object, audio and motion. In search task, we submitted 8 manual runs, which use four methods.

This paper is organized as follows: section 2 describes the system of shot boundary determination. Then the methods for story segmentation are given in section 3. In section 4, we present our four sub-systems of high-level feature extraction task. In section 5, four types of methods for search task are described. Finally, section 6 gives some conclusions.

2. Shot Boundary Detection

This year we use almost the same of TREC-11 shot segmentation system [Wu02]. FFD (*Frame-to-Frame Difference*) calculated by luminance difference and color histogram similarity are used to detect the shot changes. We use two thresholds θ_C and θ_G, which are calculated automatically according to the FFD value histogram in 500 frames, to detect if there is a clear FFD value change caused by shot changes. Then flashlight detection and motion detection are applied on candidate shot changes to remove the false alarms of cut and gradual. Fade in/out detection is applied to all candidate gradual changes. If a black screen chain exists in the candidate duration, we think it is a fade. Otherwise, it will be labeled as dissolve. The parameters used in the system are trained and adjusted based on the TRECVID2003 Development Data.

Evaluation results show that performance of our system decreases a lot this year. We found that there are a lot of short graduals which duration is smaller then 3 frames in reference answer. In our system, such short graduals will be considered as cut because they are too short. Thus, a cut insertion error and a gradual deletion error will happen so that the cut precision and gradual recall are low.

3. Story Segmentation

News story segmentation is a new task of TRECVID2003, which requires segmenting news videos into story units and classifying them into two categories, namely news and miscellaneous. Considering the structural characteristic of news videos, there are some differences between news story segmentation and general scene segmentation. The latter mainly depends on similarity-based cluster algorithms while the former one involves anchorperson detection and topic detection, etc.

3.1 Anchorpersion(AP) Detection

In news video, anchorperson shot tends to be the beginning of a news story. According to our survey, almost half of the story boundaries are anchor person shots in this year's corpus. Therefore, anchorperson shot detection is crucial for news story segmentation.

Since anchorperson shows up in a news video repeatedly, and the anchorperson shots are similar so that we can detect them by a clustering procedure described below:

1. Initialization: $GroupNum$=1, $Group_1$={$Shot_1$}, i=2.
2. For $shot_i$, calculate the similarity with all existing groups and find the $group_k$ with the maximum similarity.

$$Lum(pixel) = 0.31 pixel_{red} + 0.10 pixel_{green} + 0.59 pixel_{blue} \tag{3.1}$$

$$ShotSimilariry(Shot_i, Shot_j) = \sum_k \left| Lum_{keyframe_i}(pixel_k) - Lum_{keyframe_j}(pixel_k) \right| \tag{3.2}$$

$$GroupSimilarity(Shot_i, Group_j) = \min_{Shot_k \in Group_j} \{ShotSimilarity(Shot_i, Shot_k)\} \tag{3.3}$$

$$k = \arg\max_j \{GroupSimilarity(Shot_i, Group_j),\ j = 1, \cdots, GroupNum\} \tag{3.4}$$

3. If $GroupSimilarity(Shot_i, Group_k)$ exceeds a threshold, we add $shot_i$ into $group_k$.
   Otherwise, create a new group which consists of $shot_i$, $GroupNum = GroupNum + 1$.
4. i=i+1, goto step 2.

After the clustering procedure, all the video shots are put into several groups. To decide whether a group is an anchorperson shots group, we filter them step by step. At first, we discard the groups consisting of only one shot. Then several filters are designed to improve the performance. Since commercials also occur in TV repeatedly, they cause several false alarms. We use an advertisement filter as the first one to remove groups containing ads. The second filter used is the non-face filter. We use face detection result to remove those groups not containing face. When adopting the two filters above, we tried two strategies: strict and loose. In strict strategy, we discard the group as long as it has one commercial shot (or non-face shot) while in loose one we discard a group only when all of its shots are commercial shots (or non-face shots). Apparently, strict strategy guaranteed precision at the sacrifice of recall and vice versa. However, as long as F-value is concerned, loose strategy performed a bit better than the strict one.

The third and the fourth filters are motion filter and range filter. Since anchorperson shots are supposed to be still shots, we removed the groups with great motion. We consider a shot as a still shot if 65% of its frames are still and a group as still if 80% of its shots are still shots. We also discard the groups whose range is less than ten shots concerning the fact that anchorperson shots tend to scatter all along the video. By adopting the

latter two filters, the F-value rises several percents.

Nevertheless, in TRECVID 2003 corpus, anchorperson shots are characteristic of complex background and variant anchorperson position (see the key frames shown in Figure.1). This affects the performance of our methods. Our anchorperson detection algorithm is capable of detecting only 60% of all the anchorperson shots, which is the cause for the poor recall of our final news segmentation system.

Figure.1 Some examples: keyframes of anchorperson shot

In order to make the clustering more robust when background and anchorperson position are variant, we split the keyframe of each shot into three parts with same size: Left, Middle and Right. Equation 3.2 in clustering algorithm is replaced with:

$$ShotSimilariry(Shot_i, Shot_j) = \max_{m,n\in\{left,middle,right\}} \left\{ \sum_k \left| L_{keyframe_i,m}(pixel_k) - L_{keyframe_j,n}(pixel_k) \right| \right\} \quad (3.5)$$

However, we found that this method caused a lot of false alarms with a little improvement on recall. Finally, we still use equation 3.2 to measure the similarity between two shots.

3.2 Text Segmentation

Text segmentation has been studied for decades. However, for news story segmentation, it has to change somewhat, since what we get is not the close caption but ASR transcription. We mainly consider the text similarity at some candidate boundaries. First, we establish a vocabulary containing only nouns and verbs from the news video ASR transcription made by LIMSI [Gauvain02]. Secondly, for each candidate boundary, we build two word histograms based on certain account of words before and after this boundary (in experiments, we chose 50 words). Then, we calculate the histogram intersection as the similarity at each candidate boundary and found the valley of the similarity curves. If the similarity value of this valley is less than a threshold, it is determined to be a story boundary. There is a problem for this method: if there are story boundaries in the first or the last 50 words, they cannot be detected.

3.3 Other useful information

There are some other information which is probably useful to story boundary detection, such as commercial, speaker change, audio type change, etc. Change from non-commercial to commercial may indicate a change from news to miscellaneous, therefore creates a story boundary. Actually, a combination of anchorperson shot and commercial renders the precision decreasing and the recall increasing about ten percents compared with using anchorperson information only.

3.4 News Story Segmentation

We used several types of methods. For type A, sentence boundary and shot boundary are selected as candidate boundaries for text segment algorithm mentioned in section 3.2. After experiments, we figure it out that text segmentation at shot boundary performs better than at sentence boundary. This is because sentence boundary detection in ASR is based on silence detection, which is not so reliable. It will merge several sentences into one and result in much lower recall for our story boundary detection system. The final evaluation of our text segmentation system shows that its performance is much poorer than the methods that have been reported so far. We believe it is to some degree due to the complex structure of news text: some long stories are made up of tens of sentences while some short one consists of only one sentence.

For type B and type C, we use both audio (i.e. text) and visual features described above. Rules and maximum entropy model are selected as classifiers to decide whether a shot boundary is a news story boundary. We do not consider sub-shot because about 92% of all the story boundaries occurs at shot boundary according to our investigation on TRECVID Development data.

For type B with rules they are simple but turn out to be efficient. We consider a shot boundary to be a story boundary if it is a change from non-AP shot to AP shot or it is a change from ads to non-ads.

For type C with rules, we combine the results of type A run and type B run in intersection way and union way. Evidently, intersection way can get high precision with low recall and vice versa. In our submission, we choose the intersection run though the two F-values are of little difference.

We use maximum entropy classifier because it is efficient when the features are discrete and can be effectively selected automatically. However, maximum entropy is beaten in type B task by rules, but interestingly surpasses the rule classifier in type C that involves text information. This may be caused by the low recall of intersection operation when applying rules in type C.

Table 1 News Story Segmentation Evaluation

	Type A		Type B		Type C		Best	Median
	Shot	Sent	Rule	ME	Rule	ME		
Recall	0.587	0.270	0.500	0.427	0.258	0.378	0.790	0.378
Precision	0.272	0.287	0.584	0.644	0.825	0.773	0.844	0.584
F-Value	0.372	0.278	0.539	0.514	0.393	0.508	0.775	0.386

Table 1 shows our evaluation results. We can find that visual information is more effective than text information. Meanwhile, visual information can improve the system which only using text information. In our experiments on audio features, we found that speaker change and audio type change are not helpful. It is probably because news story boundary is not necessarily the speaker change point and music is rare in news that sometimes co-occurs with speech.

3.5 News Story Classification

In video type classification, we use the three ratios below as features:

$$Ratio_{AD} = \frac{\#\ of\ \ commercial\ \ shots\ in\ story\ i}{\#\ of\ \ shots\ in\ story\ i} \tag{3.6}$$

$$Ratio_{Speech} = \frac{\#\ of\ \ speech\ shots\ in\ story\ i}{\#\ of\ \ shots\ in\ story\ i} \tag{3.7}$$

$$Ratio_{Music} = \frac{\#\ of\ music\ shots\ in\ story\ i}{\#\ of\ shots\ in\ story\ i} \tag{3.8}$$

We use TRECVID 2003 Development data as the training data and use GMM and maximum entropy model as classifiers. From the results shown in Table 2, we can find maximum entropy model gives more satisfactory classification results.

Table 2 News Story Classification Evaluation

	Type B Rule		Type B ME		Type C ME		Best	Median
	GMM	ME	GMM	ME	GMM	ME		
Recall	0.836	0.935	0.672	0.799	0.632	0.767	1.000	0.899
Precision	0.873	0.848	0.893	0.831	0.875	0.819	0.965	0.842
F-Value	0.854	0.889	0.767	0.815	0.734	0.792	0.944	0.808

For Type A runs, only ASR transcription can be used for news story classification. We use a text classification algorithm based on VSM (Vector Space Model) [Huang01].

4. High-level Feature Extraction

We classify the high-level features into four categories:

- Scene feature: High level features concern about the scene information, such as outdoor, building, road, vegetation, non-studio setting, sport events and weather news. They can be extracted through the low-level features of the whole images.
- Object feature: High level features concern about the object in the video, such as car, aircraft, animal, face, people and person X. They can be extracted through the low-level features of the regional images.
- Audio feature: High level features relevant to audio information, such as monologue and female speech. They can be extracted from audio features.
- Motion feature: High level features relevant to camera motion, such as zoom-in and physical violence. They can be extracted from motion information.

In section 4.1, we describe a general approach to the scene feature detection and a special system for vegetation detection. In section 4.2, we introduce our methods for the object feature detection, such as face detection, car detection etc. Audio feature detection and motion feature detection are described in section 4.3 and 4.4 respectively. Finally, an ASR based method is discussed in section 4.5.

4.1 Scene Feature Detection

For the scene features such as outdoor, non-studio setting, weather news, sports event, building, vegetation and road, we collect the positive and negative training images from the TRECVID2003 Development Data with TRECVID2003 Annotation, and then use the following system to extract these features. Besides the general system, we used a special system for vegetation detection.

4.1.1 General Scene Feature Detection System

The system calculates the confidence for each key frame instead of the whole shot. Each key frame is divided into 4*4 sub-blocks and the following low-level features are extracted on these sub-blocks. There are

totally nine features used in our system.

- LAB color histogram (3 features: each feature corresponding to 64 bins for single color channels)
- Edge direction histogram (1 feature: 73 bins, the first 72 bins corresponding to the 72 edge directions and the last bin corresponding to the ratio between the non-edge pixels and total pixels)
- Co-occurrence texture: (5 features: entropy, correlation, contrast, uniformity, inverse difference moment) [Haralick73]

We calculate the confidence as a fuzzy KNN classification. For the incoming image x, we find its k nearest neighbors in training samples that are labeled $\{x_i, i = 1,2, \ldots, k\}$, the confidence is defined as follow:

$$conf(x) = \frac{\sum_{i=1}^{k} conf(x_i)\left(SIM_{total}(x, x_i)^{b-1}\right)}{\sum_{i=1}^{k} SIM_{total}(x, x_i)^{b-1}}, \quad b>1 \tag{4.1}$$

where SIM_{total} is the similarity between two images with the low-level features mentioned above, which are calculated by summing the similarities of 16 sub-blocks. $conf(x_i)$ is the confidence of training samples, which can be 1 or 0 depending on whether x_i is positive or negative. The parameter b is used to represent the importance of the similarity.

In equation 4.1, we can get the different similarities of two images with their different low-level features used. For different scene feature, different low-level feature has different discriminative abilities. Therefore, feature selection and fusion are important for the scene features detection. Simply combining every feature by experience may not get the optimal results. In this paper, we use a AdaBoost based algorithm [Freund96] to automatically adjust the features weight. The mechanism of boosting [Schapire98] is combining many weak classifiers to a final strong classifier by giving bigger weights to the better classifiers and smaller weights to the worse classifiers. It can automatically re-weighting the training samples and adjust the weights of the classifiers. In our algorithm, each weak classifier is trained use one feature. Therefore, the feature selection can be realized by adjust the weight of the respective weak classifier.

In our system, each weak classifier is a fuzzy classifier as follow:

$$h_j(x) = conf_j(x) = \frac{\sum_{i=1}^{k} conf_j(x_i)\left(SIM_j(x, x_i)^{b-1}\right)}{\sum_{i=1}^{k} SIM_j(x, x_i)^{b-1}}, b>1 \tag{4.2}$$

where, $conf_j(x)$ is the confidence calculated with *j-th* feature of the sample x. After automatic fusion of all the classifiers, we can get the optimal results. The confidence got by final classifier is used as the ranking value for each keyframe.

The variant AdaBoost algorithm is described below:

- For the training samples set (x_1,y_1) , … , (x_n,y_n) , where $y_i = 1, 0$ respective represent the positives and the negatives.
- Initialize weights $w_{1,i} = \begin{cases} 1/2m & \text{when } y_i = 1 \\ 1/2l & \text{when } y_i = 0 \end{cases}$ for training example i, where m and l are the number of positives and negatives respectively.

- Suppose we extract T low-level feature from images