CMU Informedia@TRECVID 2015 MED

Shoou-I Yu, Lu Jiang, Zhongwen Xu, Zhenzhong Lan, Shicheng Xu, Xiaojun Chang, Xuanchong Li, Zexi Mao, Chuang Gan, Yajie Miao, Xingzhong Du, Yang Cai, Lara Martin, Nikolas Wolfe, Anurag Kumar, Huan Li, Ming Lin, Zhigang Ma, Yi Yang, Deyu Meng, Shiguang Shan, Pinar Duygulu Sahin, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Teruko Mitamura, Richard Stern and Alexander Hauptmann
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh,15213

Abstract

We report on our system used in the TRECVID 2015 Multimedia Event Detection (MED) task. On the MED task, the CMU team submitted runs in the Semantic Query (SQ) and 10Ex settings. The proposed system is essentially the same as our MED 2014 system.

1.1 MED System Description

On the MED task, the CMU team uses the MED 2014 [1] system which has enabled the system to achieve good performance in the 000Ex and 010Ex settings. Furthermore, our system is very efficient in that it can complete Event Query Generation (EQG) in 16 minutes and Event Search (ES) over 200,000 videos in less than 5 minutes on a single workstation. Please see [1] for the detailed system for the details about our 000Ex and 010Ex runs.

The CMU system utilizes multiple modalities, classifiers and fusion methods to perform Multimedia Event Detection. The multiple modalities include visual, audio and text modalities. For 10Ex, two classifiers were used: linear SVM and linear regression. The fusion method used for 010Ex is the Multistage Hybrid Late Fusion, which is a combination of many different fusion algorithms. For the 000Ex runs, we utilize concept detection results from 3000 concept detectors during the SQG and ES stage.

We submitted four runs for this year's PS condition:
CMU_MED15_MED15EvalFull_PS_10Ex_MED_p-baseline_1: The baseline 10Ex system similar to our 2014 system (using same set of features).
CMU_MED15_MED15EvalFull_PS_0Ex_MED_p-expert_1: The 0Ex system using the manual queries selected by experts (using the same queries in our 2014 system).
CMU_MED15_MED15EvalFull_PS_0Ex_MED_c-autosqg_1: The 0Ex system using the automatically generated queries. The automatic query generation process is detailed in [3].
CMU_MED15_MED15EvalFull_PS_0Ex_MED_c-autosqgvisualonly_1: The 0Ex system using the automatically generated queries in [3] with only visual features (not including ASR and OCR).

1.2 Hardware Description

We utilize the following hardware for metadata generation:

1. PSC Blacklight cluster
   100 nodes, each with 4 Intel(R) Xeon(R) CPU E5620 2.40 GHz CPUs (4 cores), 128 GB RAM.
   Lustre fistributed filesystem, where we used around 50TB.
2. Rocks cluster
   20 nodes, each with 2 Intel XEON E5649 2.53 GHz CPUs (6 cores), 64 GB RAM
   4 nodes, each with 4 Intel XEON E5-2660 2.20 GHz CPUs (8 cores), 128 GB RAM.
   3 nodes, each with 4 NVIDIA TESLA K20 GPUs.
   2 data servers, 30TB each

For 10Ex event search, we use:
1 Intel(R) Xeon(R) CPU E5-2640 2.50 GHz CPU (12 cores), 128GB RAM, 4 NVIDIA TESLA K20s (2496 cores each), SSD RAID with 4TB storage.

For 0Ex event search, we use:
1 Intel(R) Xeon(R) CPU E5649 @ 2.53GHz, 64GB RAM, with a 256GB non-SSD Hard Disk.

1.3 System Performance

We report our performance on MED15EvalFull Pre-Specified Events.

Runs (MED15EvalFull)	Performance			
	MAP%	iP10	iP50	infAP200
000Ex autosqgvisualonly	6.4	0.13	0.135	0.0611
000Ex autosqg	7.8	0.2	0.188	0.1005
000Ex expert	15.1	0.38	0.307	0.2137
010Ex baseline	19.2	0.495	0.394	0.2376

Runs (MED15EvalSub)	Performance			
	MAP%	iP10	iP50	infAP200
000Ex autosqgvisualonly	9.7	0.19	0.148	0.0895
000Ex autosqg	11.0	0.245	0.174	0.1223
000Ex expert	20.6	0.39	0.285	0.246
010Ex baseline	25.5	0.515	0.343	0.2882

Acknowledgments

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, it used the Blacklight system at the Pittsburgh Supercomputing Center (PSC). This work was supported in part by the US Department of Defense, U. S. Army Research Office (W911NF-13-1-0277) and by the National Science Foundation under Grant No. IIS-1251187. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, the National Science Foundation or the U.S. Government.

References

[1] Yu, Shoou-I., et al. "Informedia@ TRECVID 2014 MED and MER." NIST TRECVID Video Retrieval Evaluation Workshop. 2014.

[2] Paul Over and George Awad and Martial Michel and Jonathan Fiscus and Wessel Kraaij and Alan F. Smeaton and Georges Quéenot and Roeland Ordelman. TRECVID 2015 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. Proceedings of TRECVID 2015, 2015

[3] Lu Jiang, Shoou-I Yu, Teruko Mitamura, Alexander Hauptmann. Bridging the Ultimate Semantic Gap: A Semantic Search Engine for Internet Videos. In ACM International Conference on Multimedia Retrieval (ICMR), 2015.

CMU Informedia @ TRECVID 2015: Semantic Indexing

Lu Jiang, Xiaojun Chang, Zexi Mao, Anil Armagan, Zhengzhong Lan, Xuanchong Li, Shoou-I Yu, Yi Yang, Deyu Meng, Pinar Duygulu-Sahin, Alexander Hauptmann

Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We report on our system used in the TRECVID 2015 Semantic Indexing (SIN) task. On the task, the CMU team submitted runs using our 2014 subsystem which only uses the improved dense trajectory features. The training set used is identical to the set used last year, which includes around 370 thousand shots from IACC.1.tv10.training and IACC.1.A-C collections. The proposed system is a subsystem of our SIN 2014 system, and thus has worse performance. The details about our 2014 system and submitted dense trajectory runs can be found in [1].

References

[1] Jiang, Lu, Xiaojun Chang, Zexi Mao, Anil Armagan, Zhengzhong Lan, Xuanchong Li, S. Yu et al. "CMU Informedia@ TRECVID 2014: Semantic Indexing." In Proc TRECVID 2014 Workshop, Gaithersburg, USA. 2014.

CMU-SMU@TRECVID 2015: Video Hyperlinking

Zhiyong Cheng[1], Xuanchong Li[2], Jialie Shen[1], and Alexander G. Hauptmann[2]

[1]Singapore Management University,Singapore 178902
[2]Carnegie Mellon University, Pittsburgh, PA, 15213 ,
{zy.cheng.2011,jlshen}@smu.edu.sg, {xcli,alex}@cs.cmu.edu

Abstract

In this report, we describe CMU-SMU's participation in the Video Hyperlinking task of TRECVID 2015. We treat video hyperlinking as ad-hoc retrieval scenario and use a variety of retrieval methods. Our experiments mainly focus on the study of different features on the performance of video hyperlinking, including subtitle, metadata, audio and visual features, as well as the consideration of surrounding context. Different combination strategies are used to combine those features. Besides, we also attempt to categorize the queries and use different search strategies for different categories. Experiments results show that (1) the context does not generally improve results, (2) the search performance mainly rely on textual features, and the combination of audio and visual feature cannot provide improvements; (3) due to the lack of training examples, machine learning techniques cannot provide contributions.

1 Introduction

With the explosive growth and the widespread accessibility of multimedia content on the Web, video content is becoming one of the most valuable sources to assess information and knowledge [9, 36]. In the consumer of video content, it is common that users are interested to find further information on some aspects of the topic of interest contained within a video segment. Therefore, it is crucial to develop effective video search and hyperlinking to help users explore, navigate and search interest video contents in audiovisual archives. Video hyperlinking is to link a video anchor or segment to other video segments in a video collection, based on similarity or relatedness. Accordingly, video hyperlinking enables users to navigate between video segment in a source content and related elements in the same content file [3, 8, 16].

To facilitate the development and advancement of video hyperlinking system, video hyperlinking has becoming a competition task since 2012 in MediaEval [14, 15, 17]. Standard test collections are provided and evaluation metrics are defined for the evaluation of developed systems. The task is defined to find *relevant anchors* or *short segments* (e.g., 2 minutes) of video contents given a set of query anchors. Thus, the hyperlinking is generally addressed within an information retrieval framework. As the videos in test collection could be in hours of length, video hyperlinking consists of two steps: (1) *video segmentation* - separate a video into a number of clips and (2) *video retrieval* - retrieval potential links to video or video segments. [1] Many systems apply the fixed-length segmentation method to separate videos into fixed-length of segments [11, 31, 19, 30]. Other video segmentation methods were also developed and studied, such as video shot based [10, 33] and semantic-based segmentation [7, 12, 23]. More efforts have been development of effective retrieval methods, including the exploration of different source information (e.g., subtitle, metadata, transcriptions, segment surrounding context [19], name entity [13, 31, 11], enrichment of concept

[1]The order of the two steps can be reverse, firstly retrieving potential relevant video and then extracting the most relevant segments from the video identified in the first step [22, 16].

and synonyms [30, 33], as well as audio [19] and visual features [10, 30, 19]) and search strategies (e.g., combination with or re-ranking with visual features [7, 30, 6], combination of video-level and segment-level retrieval [11], etc.).

In this paper, we report our participants in the TRECVID 2015 Video Hyperlinking Task. We use the fixed-length video segmentation method and focus on studying the effects of different types of information sources on the performance of video hyperlinking, including text (subtitle, metadata, transcription) and a variety of video content (audio, visual and motion) features. Nine different text-based retrieval methods are used based on the text information with and without the consideration of surrounding context (around the query or target segment). Besides, we also study the performance of multimodal feature combination using weighted linear combination and learning-to-ranking methods. Further, we attempt to classify the query anchors into different categories and using different combination weights for different categories. Experiments on the development set show that surrounding context and video-content features have little contribution on the performance improvement.

2 Video Hyperlinking

In this section, we describe the dataset, the specific task of video hyperlinking and evaluation metrics used in our experiments.

2.1 Description of Task

The video hyperlinking task is to find video segments which contain relevant or supplemental information to a given query segment in the video collection. The formal definition of the video hyperlinking task in TRECVID 2015 is: *given a set of test videos with metadata with a defined set of anchors, each defined by start time and end time in the video, return for each anchor a ranked list of hyperlinking targets: video segments defined by a video ID and start time and end time*. In evaluation, a ranking list of 1000 link targets for each test query anchor. Hyperlinking targets pointing to the video where the anchor was extracted from should be excluded and will be disregarded during the evaluation, namely, the returned video segment and query anchor should from different videos (Notice that the duration of a video can be up to 10 hours, and the duration of a query anchor and returned anchors/segments are usually 10 to 120 seconds).

2.2 Dataset

The dataset consists of 2500-3500 hours of BBC video content. The data is accompanied with metadata (title, short program descriptions and subtitles), automatic speech recognition (ASR) transcripts by LIMSI [20, 27], LIUM [34] and NST-Sheffield [24, 29], two versions of concept detectors, as well as prosodic audio features [18]. To facilitate the development of video hyperlinking systems, a develop set of 30 query anchors with a set of ground-truth anchors are provided. The number of positive examples for the development query anchors varies from 17 to 122. Notice that many of the positive examples are from the same video where the corresponding query anchors was extracted from. Details about the query anchors and ground-truth in the development set is shown in Table 1. 135 test query anchors are provided for the final evaluation of the designed systems.

Table 1: Development set of query anchors and corresponding results.

# Query	Duration (s)			# Positive Results		
	Min	Max	Mean (Std.)	Min	Max	Mean (Std.)
30	3	183	22.97($\pm$33.21)	17	122	62.93($\pm$26.97)

2.3 Evaluation

To evaluate the performance of video hyperlinking systems, top ranking results of submissions are accessed using a mechanical turk (MT) crowd-sourcing approach, assessing the top ranked documents. A test assessment on a smaller part of the data by a local team of target users is used to identify potential discrepancies between the MT workers' judgments and those of the target user group. Descriptions given by the anchor creators (anchor descriptions, description and format requested targets) are used for evaluation purpose. In the generation of ground truth, only a subset of the submissions for each query will be used in evaluation. To reduce the workload of evaluators, for the anchors longer than 2 minutes, only the first two minutes will be used as the basis of relevance assessment. For more details about hyperlinking evaluation, please refer to [16].

The submissions are evaluated based on the precision at a certain rank measure, adapted to unconstrained time segments. In this paper, we report the performance on evaluation metrics of Precision@$\{5, 10, 20\}$, MAP, MAP_bin, and MAP_tol. Please refer to [2] for the descriptions about the evaluation metrics.

3 Video Hyperlinking System

We addressed the video hyperlinking as an ad-hoc retrieval problem. Given a query anchor indexed with certain features, video segments in the test collection are also indexed with the same feature and method, and then a retrieval method is used to search and return the most relevant video segments for this query. In our experiments, we (1) first separate each video in the collections into *50s fixed-length* segments without overlapping, as the use of 50s length segments has obtained good performance in CUNI2014 video hyperlinking system [19]; (2) from each segment, different types of features are extracted and indexed for retrieval; (3) for the extracted features, a variety of retrieval methods are explored; and (4) different strategies are used to combine the results obtained based on different features. In the next, we describe the used features and retrieval methods in experiments.

3.1 Retrieval Methods

3.1.1 Text-based Method

Text Features. We explore the effectiveness of different sources of textual information in video hyperlinking, including subtitle and three types of transcriptions (LIMSI, LIUM, and NST-Sheffield). For each type of the feature, we also consider their combination with metadata as well as surrounding contexts. The tested lengths of surrounding segments include 50s, 100s, and 200s. Accordingly, for each of subtitle, LIMSI, LIUM and NST-Sheffield, there are eight indexing methods. Taken *subtitle* as an example, there are *subtitle*, *subtitle with 50s context*, *subtitle with 100s context*, *subtitle with 200s context*, *subtitle and metadata*, *subtitle and metadata with 50s context*, *subtitle and metadata with 100s context* and *subtitle and metadata with 200s context*. For a segment, *subtitle and metadata* is to concatenate the subtitle of this segment with the metadata of the video from which the segment is extracted. Similarly, *subtitle and metadata with 50s context* are the concatenation of the subtitle of this segment and 50-seconds-length passage before and after the segment and the metadata of the corresponding video. All the textual resources are preprocessed by removing punctuation, normalizing capitalization and removing stop words.

Retrieval Methods. For each type of features, we experimented with nine different retrieval models: (1) BM25, (2) DFR version of BM25(DFR-BM25) [21], (3) DLH hyper-geometric DFR model (DLH13) [4], (4) DPH [5], (5) Hiemastra's Language Model (Hiemastra-LM) [26], (6) InL2 - inverse document frequency model for randomness, Laplace succession for first normalisation, and normalisation 2 for term frequency normalisation [21], (7) TF-IDF, (8) LemurTF-IDF [1], and (9) PL2 - poisson estimation for randomness, Laplace succession for first normalisation, and normalisation 2 for term frequency normalisation [21]. We used Terrier[2] IR system to run experiments with these retrieval methods (with default parameters) with different textual sources.

3.1.2 Content-based Method

For the content-based method, we use various video features, including motion feature, audio feature, semantics feature, etc., to do the retrieval task. We also employ the Learning to Rank [25] technique to do the result fusion.

Video Features.

[2]http://www.terrier.org

- Motion Feature: CMU Improved Dense Trajectory [28]: 3 different versions.
- Audio Feature: MFCC: 2 different versions.
- Visual Semantic Feature [32]: 6 different versions.

Retrieval Methods. For each video feature, we use the simple linear distance to compute the relevance score. A problem is that the feature might not work well in a linear space. We remedy the problem by using the explicit feature map [35]. It approximates the non-linear space by an explicit feature mapping. Finally, we use learning to rank methods to fuse the features together.

3.1.3 Multimodal-based Method

We explore the effects of the combination of different features in video hyperlinking, based on the assumption that different features could capture different aspects of a video segment.

Weighted Linear Combination (WLC). In this method, the relevant score of a video segment with respect to a query is computed by a weighted linear combination of the relevant scores obtained by different features. Let $wlc(q, v)$ is the final relevance score obtained by the weighted linear combination, and $rel(f_i)$ is the relevance score obtained based on feature f_i. Given the selected feature $\{f_1, f_2, \cdots, f_n\}$, the $wlc(q, v)$ is computed by:

$$wlc(q, v) = w_1 \cdot rel(f_1) + w_2 \cdot rel(f_2) + \cdots + w_n \cdot rel(f_n) \quad (1)$$

where $\boldsymbol{w} = \{w_1, w_2, \cdots, w_n\}$, the $wlc(q, v)$ is the linear combination weights, which characterize the contribution of different features on the final performance. The training set is to learn the optimal weight $\boldsymbol{w}$. Due to the few training examples, we only used 6 features in our experiments. These features are selected based on on their individual performances and the consideration of combining heterogeneous features. Specifically, the selected features are: Subtitle_Metadata_LemurTF-IDF, Subtitle_Metadata_DPH, Key_Concept_TF-IDF, improved trajectory and MFCC. Subtitle_Metadata_LemurTF-IDF denotes that the relevant score is obtained by LemurTF-IDF based on the subtitle and metatada. Similar definition is applied for other methods. Key_Concept_LemurTF-IDF using the TF-IDF method based on the key concepts of keyframes learned by the Leuven method. For a video segment, the key concepts of all the frames in this segment are concatenated together to form its key concept representation.

For different types of videos, their contents or topics could be very different. The contributions of features for different types of video contents in hyperlinking could be very different. Thus, it would be useful to using different weights for different video categories. Accordingly, we classify the videos into categories based on the programme category ontology of BBC news[3]. Due to the limited query examples in the development dataset, we further group the videos into two broad categories:

- Category 1: news & weather; science & nature; music (religion & ethics); travel; politics news; life stories music; sport (tennis); food & drink; motosport.
- Category 2: history; arts, culture & the media; comedy (sitcoms), cars & motors; antiques, homes & garden, pets & animals; health & wellbeing, beauty & style.

In general, videos in the sub-categories of Category 1 enjoy more similar contents in text, audio and visual features (such as news and music), and thus queries in Category 1 are easier to get better results. In contrast, for videos in the same sub-categories in Category 2, although their contents are about the same topic, but the contents could be very different in contents. For example, videos about history or health could be very different in words and scenes. To evaluate the performance of this method, we randomly split the query anchors in development set into training set and test set. The details of training dataset and test dataset for global weighted linearly combination (GWLC - without the consideration of video categories) and categorized weighted linear combination (CWLC) are described in Table 2. Notice that the training example is very limited, especially for CWLC method, which limits the performance of the weighted linear combination.

Learning to Rank is a method that applying the machine learning on the retrieval, which can refine the retrieval results. In this task, we use the retrieval scores from the various feature as the input of

[3]The videos can be categorized based on the name of the video based on the programme categories in BBC, such as "bbctwo_the_daily_politics" is in the category of *politics news*.

Table 2: Sizes of training set and test set in global weighted linear combination (whole) and categorized weighted linear combination (category 1 and category 2).

	# queries in training set	# queries in testing set
whole	15	15
category 1	9	9
category 2	8	4

the learning algorithms (such as linear regression, naive bayes, SVM, etc.). The output is regarded as the final retrieval scores.

4 Experimental Results

4.1 Performance on Development Data

In this section, we report the experiment results of different methods on the development dataset. The results of content-based methods have not presented because of the overall poor performance.

Table 3: Results of the Hyperlinking task for different transcripts, metadata, retrieval methods and contexts. In each row, the retrieval method is the best retrieval methods among the nine tested methods for the corresponding text source. Please refer to Sect. 3.1.1 for the retrieval method in the "Method" column: (1) BM25, (3) DLH13, (4) DHP, (5) Hiemastra-LM, (8) LemurTF-IDF, and (9) PL2. NST refers to NST-Sheffield transcript.

Transcripts	Metadata	Context	Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
Subtitle	No	No	(8)	**.1622**	**.3241**	**.2966**	**.2276**	**.1037**	**.0798**
LIMSI	No	No	(8)	.0928	.2154	.1731	.1365	.0581	.0419
LIUM	No	No	(1)	.0557	.1440	.1240	.0980	.0464	.0278
NST	No	No	(8)	.0650	.1643	.1286	.1018	.0488	.0323
Subtitle	Yes	No	(8)	**.1971**	**.2933**	**.2533**	**.2050**	**.1107**	**.0692**
LIMSI	Yes	No	(8)	.1464	.2000	.1733	.1467	.0863	.0493
LIUM	Yes	No	(4)	.1069	.1467	.1567	.1317	.0672	.0333
NST	Yes	No	(8)	.1229	.1533	.1467	.1283	.0776	.0420
Subtitle	No	50s	(9)	.1144	.1733	.1367	.1183	.0587	.0255
Subtitle	No	100s	(5)	.1236	.2200	.1700	.1317	.0560	.0314
Subtitle	No	200s	(3)	.1279	.2267	.1600	.1033	.0550	.0339
Subtitle	Yes	50s	(3)	.1243	.2000	.1467	.1117	.0641	.0288
Subtitle	Yes	100s	(5)	**.1362**	.2200	.1800	**.1350**	**.0680**	.0327
Subttile	Yes	200s	(3)	.1343	**.2467**	**.1939**	.1133	.0577	**.0362**

4.1.1 Text-based Retrieval Method

The results of text-based retrieval methods using different text sources are presented in Table 3. For each text source, only the best performance obtained by the nine retrieval methods is reported. As a large set of text-based retrieval methods (different text sources and different retrieval methods) has been explored, we have not presented the results of all methods. The results are grouped into three groups in the table. As the performance of using subtitle is much better than the use of ASR transcripts (LIMSI, LIUM and NST-Sheffield), we did not show the performance of automatic generated transcripts with the consideration of context. The performance based on ASR transcripts is limited by the speech recognition accuracy. Among the three ASR transcripts, LIMSI obtains the best performance, followed by LIUM. Not surprising, with the consideration of metadata, the performance of ASR transcripts can be significant improved, as the metadata is manually annotated and summarizes the video contents. Comparing to only using subtitle, the combination of metadata improves the performance on MAP, while the precisions on top results have been decreased. As metadata contains the summary of a video, it could lead to retrieve a video segment which is in a video with the same topic as the video of the query segment, while the video segment is irrelevant to the query segment. While for a video segment which is relevant to the query segment, if the topic of the corresponding videos, the consideration of metadata could increase the relevance score and thus move the video segment to higher position in the result ranking list, leading to the increase of MAP.

From the results of the third group in the table, the consideration of context data cause the performance significantly decreased. The results imply that the incorporation of context data introduce noisy data, which mislead the search of relevant segment. By comparing the search methods of different text sources, it can be found that better performances are obtained by vector space (LemurTF-IDF) method for text information without context (relatively short documents), and better performances are obtained by probabilistic methods with the consideration of contexts (relatively long documents).

4.1.2 Weighted Linear Combination.

Table 4 reports the performance of weighted linearly combination methods. Because the performances of different queries varied in large ranges, we list the corresponding performance of the test queries using Subtitle_Metadata_LemurTF-IDF for comparisons. It is easy to find that queries from Category 1 obtained much better results than queries from Category 2. By comparing with the performance of weighted linear combination methods, it can be seen that the performance decreases with the combination of other features based on the simple late fusion method.

Table 4: Results of the Hyperlinking task using weighted linear combination methods.

Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
LemurTF-IDF	.3054	.3692	.3385	.2808	.1514	.0992
GWLC	.2699	.4000	.3769	.3269	.1344	.0960
LemurTF-IDF (category 1)	.4324	.4667	.4556	.3833	.2075	.1373
CWLC (category 1)	.3814	.5111	.4889	.4444	.1826	.1317
LemurTF-IDF (category 2)	.0195	.1500	.0750	.0500	.0253	.0133
CWLC (category 2)	.0200	.1500	.1000	.0625	.0255	.0160

4.1.3 Performance of Multimodality Fusion

Figure 1 shows the ROC of learning to rank fusion on development data with different feature groups.

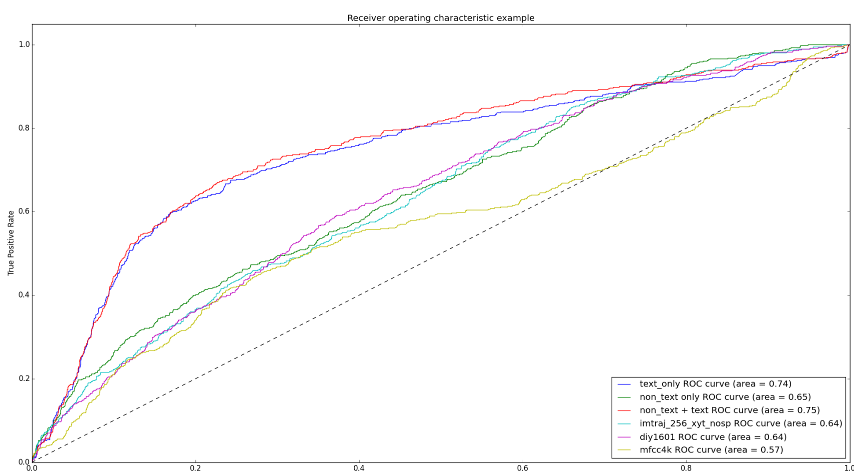

Figure 1: The ROC with Different Features on Development Dataset

A potential problem with the model is the imbalance data. In the training set, the positive/negative ratio is much higher than the testing set (real world case). The method we use is to use prior to manually correct the positive/negative ratio. An example is using the Naive Bayes with a prior that strongly set preference on negative data.

4.2 Submissions and Performance on Test Data

We submitted four runs based on each of the following methods: (1) Subtitle_Metadata_LemurTF-IDF (tv15lnk_cmu_L_4_F_M_M_LemurTFIDF), (2) Global Weighted Linearly Combination (tv15lnk_cmu_L_2_F_M_M_Fusion), (3) Categorized Weighted Linearly Combination (tv15lnk_cmu_L_3_F_M_M_CategorizedFusion), (4) Using learning to rank to fuse the best two text feature with Ridge Regression, (5) Using learning to rank to fuse the best two text feature with Naive Bayes, where the prior is strongly biased to negative. The performance of the submitted runs (after cleaning) on the test data is shown in Table 5.

Table 5: Performance of submission runs on test query set in the Hyperlinking task.

Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
L_4_F_M_M_LemurTFIDF	.4623	.6540	.6080	.4380	.2876	.2694
L_2_F_M_M_Fusion	.3159	.6300	.5340	.4025	.2813	.2440
L_3_F_M_M_CategorizedFusion	.3134	.6300	.5240	.4005	.2799	.2416
L_1_F_M_M_good.two.text.nb	.4079	.6100	.5540	.4010	.2756	.2549
L_1_F_IMSU_M_good_text_feat_ridge_test	.2301	.4040	.3880	.2715	.1752	.1560

5 Conclusion

In this notebook paper, we report our experiments in the TRECVID 2015 Video Hyperlinking task. A large set of textual and video content features on the performance of video hyperlinking has been studies. The results show that the video hyperlinking performance relies on manual annotations (subtitle and metadata). The performance based on the ASR transcriptions is still far from the performance of manual annotations, while it is much better than audio, visual and motion features. The combination of surrounding context information will decrease the performance. The use of video-content based features (audio, visual and motion) has little effects on the performance of textual features. Further, due to the lack of well-labeled data, it is difficult to use machine learning techniques to improve the performance.

Acknowledgments

This research is supported by Singapore Ministry of Education under Academic Research Fund Tier-2 (MOE Ref: MOE2013-T2-2-156).

References

[1] J. Allan, J. Callan, K. Collins-Thompson, B. Croft, F. Feng, D. Fisher, J. Lafferty, L. Larkey, T. Truong, P. Ogilvie, et al. The lemur toolkit for language modeling and information retrieval. *The Lemur Project.[WWW document] http://lemurproject. org (accessed 25 January 2012)*, 2003.

[2] R. Aly, M. Eskevich, R. Ordelman, and G. Jones. Adapting binary information retrieval evaluation metrics for segment-based retrieval tasks. *arXiv preprint arXiv:1312.1913*, 2013.

[3] R. Aly, R. Ordelman, M. Eskevich, G. Jones, and S. Chen. Linking inside a video collection: what and how to measure? In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 457–460, 2013.

[4] G. Amati. Frequentist and bayesian approach to information retrieval. In *Advances in Information Retrieval*, pages 13–24. 2006.

[5] G. Amati, G. Amodeo, M. Bianchi, Ca. Gaibisso, and G. Gambosi. Fub, iasi-cnr and university of tor vergata at trec 2008 blog track. In *Proceedings of the 16th Text REtrieval Conference (TREC-2007)*.

[6] Werner Bailer and Harald Stiegler. Jrs at search and hyperlinking of television content task. 2014.

[7] C. Bhatt, N. Pappas, M. Habibi, and A. Popescu-Belis. Idiap at mediaeval 2013: Search and hyperlinking task. In *Proceedings of the MediaEval Workshop*, 2013.

[8] C. Bhatt, N. Pappas, M. Habibi, and A. Popescu-Belis. Multimodal reranking of content-based recommendations for hyperlinking video snippets. In *Proceedings of International Conference on Multimedia Retrieval*, 2014.

[9] X. Chang, Y. Yu, Y. Yang, and A. Hauptmann. Searching persuasively: Joint event detection and evidence recounting with limited supervision. In *Proceedings of ACM International Conference on Multimedia*, 2015.

[10] S. Chen, G. Jones, and N. O'connor. Dcu linking runs at mediaeval 2012: search and hyperlinking task. In *Proceedings of MediaEval Workshop*, 2012.

[11] S. Chen, G. Jones, and N. O'connor. Dcu linking runs at mediaeval 2013: search and hyperlinking task. In *Proceedings of MediaEval Workshop*, 2013.

[12] S. Chen, G. Jones, and N. O'connor. Dcu linking runs at mediaeval 2014: search and hyperlinking task. In *Proceedings of MediaEval Workshop*, 2014.

[13] T. De Nies, P. Debevere, D. Van Deursen, W. De Neve, E. Mannens, and R. Van de Walle. Ghent university-ibbt at mediaeval 2012 search and hyperlinking: Semantic similarity using named entities. In *Proceedings of the MediaEval Workshop*, 2012.

[14] M. Eskevich, R. Aly, D. Racca, R. Ordelman, S. Chen, and G. Jones. The search and hyperlinking task at mediaeval 2014. In *Proceedings of MediaEval Workshop*, 2014.

[15] M. Eskevich, J. Gareth, , S. Chen, R. Aly, and R. Ordelman. Search and hyperlinking task at mediaeval 2012. In *Proceedings of MediaEval Workshop*, 2013.

[16] M. Eskevich, G. Jones, and R. Aly. Multimedia information seeking through search and hyperlinking. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 287–294, 2013.

[17] M. Eskevich, G. Jones, S. Chen, R. Aly, R. Ordelman, and M. Larson. Search and hyperlinking task at mediaeval 2012. In *Proceedings of MediaEval Workshop*, 2013.

[18] F. Eyben, F. Weninger, F. Gross, and B. Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838, 2013.

[19] P. Galušcáková, P. Pecina, M. Kruliš, and J. Lokoc. Cuni at mediaeval 2014 search and hyperlinking task: visual and prosodic features in hyperlinking. 2014.

[20] J. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. *Speech communication*, 37(1):89–108, 2002.

[21] A. Gianni. Probabilistic models for information retrieval based on divergence from randomness. *PhD Thesis, School of Computing Science, University of Glasgow*, 2003.

[22] C. Guinaudeau, G. Gravier, and P. Sébillot. Irisa at mediaeval 2012: Search and hyperlinking task. In *Proceedings of the MediaEval Workshop*, 2012.

[23] C. Guinaudeau, A. Simon, G. Gravier, and P. Sébillot. Hits and irisa at mediaeval 2013: Search and hyperlinking task. In *Proceedings of the MediaEval Workshop*, 2013.

[24] T. Hain, A. El Hannani, S. Wrigley, and V. Wan. Automatic speech recognition for scientific purposes-webasr. In *Interspeech*, pages 504–507, 2008.

[25] LI Hang. A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems*, 94(10):1854–1862, 2011.

[26] D. Hiemstra. Using language models for information retrieval. *PhD Thesis, University of Twente*, 2001.

[27] L. Lamel. Multilingual speech processing activities in quaero: Application to multimedia search in unstructured data. In *Huamn Language Technologies = The Baltic Perspective: Proceedings of the Fifth International Conference Baltic HLT*, pages 1–8, 2012.

[28] Zhenzhong Lan, Ming Lin, Xuanchong Li, Alexander G Hauptmann, and Bhiksha Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. *arXiv preprint arXiv:1411.6660*, 2014.

[29] P. Lanchantin, P. Bell, M. Gales, T. Hain, X. Liu, Y. Long, J. Quinnell, S. Renals, O. Saz, M. Seigel, S. Swietojanski, and P. Woodland. Automatic transcription of multi-genre media archives. In *First Workshop on Speech, Language and Audio in Multimedia*.

[30] M. Lokaj, H. Stiegler, and W. Bailer. Tosca-mp at search and hyperlinking of television content task. In *Proceedings of the MediaEval Workshop*, 2013.

[31] T. Nies, W. Neve, E. Mannens, and R. de Walle. Ghent university-iminds at mediaeval 2013: An unsupervised named entity-based similarity measure for search and hyperlinking. In *Proceedings of the MediaEval Workshop*, 2013.

[32] Paul Over, George M Awad, Jon Fiscus, Brian Antonishek, Martial Michel, Alan F Smeaton, Wessel Kraaij, and Georges Quénot. Trecvid 2010–an overview of the goals, tasks, data, evaluation mechanisms, and metrics. 2011.

[33] Z. Paróczi, B. Fodor, and G. Szucs. Dclab at mediaeval2014 search and hyperlinking task. 2014.

[34] A. Rousseau, P. Deléglise, and Y. Estève. Enhancing the ted-lium corpus with selected data for language modeling and more ted talks. In *The 9th edition of the Language Resources and Evaluation Conference*, pages 3935–3939, 2014.

[35] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.

[36] S. Yu, L. Jiang, Z. Xu, Y. Yang, and A. Hauptmann. Content-based video search over 1 million videos with 1 core in 1 second. In *Proceedings of ACM International Conference on Multimedia Retrieval*, 2015.

WARD-CMU@TRECVID 2015 Surveillance Event Detection

Xingzhong Du[1], Xuanchong Li[2], Xiaofang Zhou[1], Alexander Hauptmann[2]
[1]The University of Queensland, [2]Carnegie Mellon University

I. Abstract

We present a retrospective system for event detection in surveillance videos automatically, which is built on the Gatwick development data. It is an enhanced version of the retrospective system in [1]. The changes come from four aspects. First, dense trajectory [2] and improved dense trajectory [3] are used together in the proposed system. Second, the PCA features used in Gaussian Mixture Model are changed to whiten-PCA features. Third, we implement a learning-based probability function for LIBLINEAR [4]. Forth, instead of averaging all the detection scores we have, we select two kinds of them to get better results. We think all the changes are beneficial to the final submission 'WARD-CMU p-fusion_1' which wins 4 events in all 7 events. Specifically, it is worth noting that the PersonRuns sets a new record in recent years' SED competitions. Through the results in our internal evaluation, we think dense trajectory and improved dense trajectory are complementary for event detection in a complex surveillance environment. We also notice that current system is bad at detecting the short events like CellToEar, ObjectPut and Pointing. We are considering to introduce other methods such as pose estimation and pedestrian detection to enhance current system in the future.

II. Retrospective System

The retrospective system consists of five components : video preprocessing, feature extraction, feature encoding, model training and score fusion. In this section, we will briefly introduce their functionalities and our changes in them.

A. Video preprocessing

The video preprocessing prepares the input for the feature extraction. In the first step, all the videos are resized to 320×240. This accelerates the feature extraction. After that, the resized videos are slid into video clips. Specially, each clip's length is 60 frames and 30-frame overlapped with the adjacent clips. It is worth noting that the dense trajectory and improved dense trajectory track the feature points for 15 frames by default. It means that, if we want to extract all the feature points in a clip, we need the clip's length to be 75 frames. Therefore, for each video clip, we append 15 subsequent frames behind the original 60 frames. After the sliding, roughly 350k clips are generated. The clips whose size are 0 are removed at the end of video preprocessing.

B. Feature Extraction

The feature extraction processes video clips and generates the raw features. Last year, we used improved dense trajectory only. This feature can capture the wrapped object motion from the video, which is achieved by removing camera motion through homography plus RANSAC [6]. We check the released code[1], and find that improved dense trajectory removes the dominant motion between two frames actually [7]. In SED, there is no camera motion in the videos. The dominant motion comes from the stream of people. Applying improved dense trajectory in this situation can remove the interference from the irrelevant persons. That's why we used improved dense trajectory in last year's SED.

However, after last year's SED, we find that some positive clips only contain a few persons. In this situation, the dense trajectory is more suitable because removing the dominant motion is unnecessary. Therefore, the features in use for this year's SED are dense trajectory and improved dense trajectory.

C. Feature Encoding

The feature encoding encodes the raw feature array into a vector for each clip. In event detection, this step is necessary because it makes the feature representation more robust. The state-of-the-art encoding method is the spatial-temporal fisher [8]. This method consists of four steps. The first step learns a projection matrix based the raw features by Principle Components Analysis (PCA). After dimension reduction, the dimension of the raw features are reduced by a factor of two. The second step learns a Gaussian Mixture Model (GMM) based on the reduced features. The components in GMM act as the visual words for inferring the soft assignment information. The third step transforms the reduced features of a clip into fisher vectors by the soft assignment information, and averages them into one fisher vector for this clip. The last but not the least step is the normalization. It enhances the class-specific information by power and l_2 normalization [10].

In recent literature [9], the whiten PCA has been proved to be superior to PCA for action recognition. The whiten PCA rescales the reduced features to make them have similar variances. It could make the distribution of GMM components more uniform, so that the fisher vector can discriminate more patterns. Therefore, in this year's system, we use whiten PCA instead of PCA in feature encoding.

D. Model Training

The model training creates detectors for each event under Camera 1, 2, 3, 5. It consists of three steps. In the first step, we treat the clips which have 50% overlap with the ground truth as the positive, then use LIBLINEAR [4] to train detectors and two-fold cross-validation to choose parameters. Using

[1]https://lear.inrialpes.fr/people/wang/improved_trajectories

LIBLINEAR has two reasons. First, the features for learning are all fisher vectors which are suitable for linear kernels naturally. Second, the dimension of the fisher vector in use is 116736. It requires 0.5MB space of each vector on disk. linear SVM avoids storing the support vectors, which saves a lot of storage space for a detector. After the detectors are trained, the decision values from the detectors need to be transformed into probabilities, so that the decision values from different models are comparable. In LIBSVM [5] python version, we can use its packaged probability function. But in LIBLINEAR, we need to implement the probability function by ourselves. In last year's submission, the probability function is simply implemented by curve fitting. In this year's submission, the probability function is implemented as [11], which is more robust than the curve fitting. The python code can be downloaded from *https://github.com/domainxz/pytools.git*. We verify this code by reproducing the action recognition experiment in [3]. The results show this code can work properly. The third step of model training learns a threshold for each detector, then applies Non-Maximum Suppression (NMS) to merge the adjacent positive clips. When the model training is finished, We will have 7×4 detectors per feature. Each of them only focuses on one event under one camera.

E. Score Fusion

TABLE I. EVALUATIONS FOR FUSION STRATEGY

event	idtwfv	dtfv+idtwfv	idtfv+idtwfv	dtwfv+idtwfv
CellToEar	1.0058	**1.0013**	1.0036	1.0040
Embrace	1.0068	0.9253	0.9197	**0.9105**
ObjectPut	1.0042	1.0023	1.0026	**1.0020**
PeopleMeet	0.9520	**0.9238**	0.9369	0.9297
PeopleSplitUp	0.9613	0.8931	0.9036	**0.8861**
PersonRuns	0.6440	0.6478	0.6549	**0.6299**
Pointing	1.0140	0.9920	0.9891	**0.9858**

The score fusion averagely fuses the probabilities from different selected features to form the final submission. Therefore, the key problem is which features are fused together. After evaluating the detectors on the development set, we get four group of detection scores at hand. They are predicted by the fisher vectors in terms of dense trajectory with normal PCA (dtfv), dense trajectory with whiten PCA (dfwfv), improved dense trajectory with normal PCA (idtfv) and improved dense trajectory with whiten PCA (idtwfv). We try all the possible combinations to select the best, and find the combination of dtwfv and idtwfv is the best.

III. RESULT ANALYSIS

TABLE II. COMPARISON BETWEEN OUR RESULTS AND OTHERS' BEST RESULTS

Event	Our retro results		Best retro results		Best inter results	
	aDCR	mDCR	aDCR	mDCR	aDCR	mDCR
CellToEar	**1.0046**	**1.0006**	1.3071	1.0006	2.1010	1.0006
Embrace	0.8680	0.8453	0.7909	0.7909	0.8540	0.8540
ObjectPut	1.0160	0.9884	1.0120	0.9965	0.9930	0.9867
PeopleMeet	**0.8939**	**0.8848**	1.0426	0.9981	0.9978	0.9919
PeopleSplitUp	**0.8934**	**0.8785**	0.9387	0.9253	0.9164	0.9164
PersonRuns	**0.5768**	**0.5466**	0.9700	0.9545	0.9411	0.9411
Pointing	1.0140	0.9940	1.0040	0.9989	0.9939	0.9939

In this section, we firstly compare our retrospective submission to the others' best results in terms of actual DCR (aDCR) and min DCR (mDCR) by Table II. We make the event names

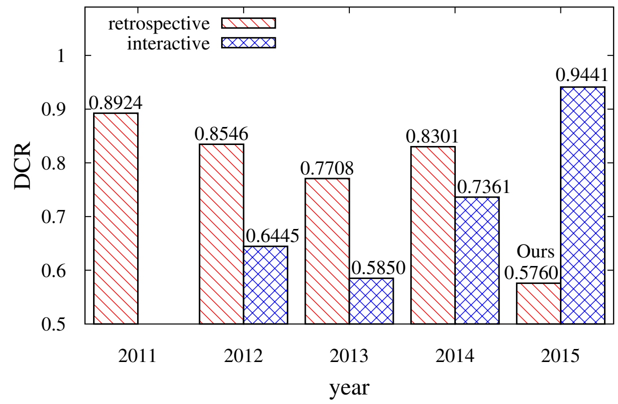

Fig. 1. The accuracy evaluated in terms of actual DCR for PersonRuns event in recent five years' SED retrospective and interactive competitions.

bold which we get the first positions in SED 2015 [12]. In total, we win 4 events in no matter the retrospective or the interactive competitions. But we notice that our method is only good at handling the events driven by long duration actions.

In this year, we achieve the lowest DCR in PersonRuns event. We compare this result to recent five years' best results in Fig. 1. We find this score achieves the new record in recent years' SED competitions, even though the test data have been changed since 2014.

IV. ACKNOWLEDGMENT

This work was supported in part by the US Department of Defense, U. S. Army Research Office (W911NF-13-1-0277) and by the National Science Foundation under Grant No. IIS-1251187. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, the National Science Foundation or the U.S. Government.

REFERENCES

[1] S.-I. Yu, L. Jiang, Z. Mao, X. Chang, X. Du, C. Gan, Z. Lan, Z. Xu, X. Li, Y. Cai *et al.*, "Informedia @ TRECVID 2014," in *NIST TRECVID Video Retrieval Evaluation Workshop*, 2014.

[2] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011, pp. 3169–3176.

[3] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013, pp. 3551–3558.

[4] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.

[5] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *TIST*, vol. 2, no. 3, p. 27, 2011.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *SCIA*, 2003, pp. 363–370.

[8] J. Krapac, J. J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *ICCV*, 2011, pp. 1487–1494.

[9] X. Peng, L. Wang, X. Wang, and Y. Qiao, “Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice,” *CoRR*, vol. abs/1405.4506, 2014.

[10] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *ECCV*, 2010, pp. 143–156.

[11] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[12] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quenot, and R. Ordelman, “Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics,” in *Proceedings of TRECVID 2015*. NIST, USA, 2015.