Informedia @ Trecvid 2016

Junwei Liang[1], Jia Chen[1], Poyao Huang[1], Xuanchong Li[1], Lu Jiang[1], Zhenzhong Lan[1], Pingbo Pan[2], Hehe Fan[2],Qin Jin[3], Jiande Sun[4], Yang Chen[5], Yi Yang[2], Alexandar Hauptmann[1]

Carnegie Mellon University[1]
University of Technology Sydney[2]
Renmin University of China[3]
Shandong University[4]
Zhejiang University[5]

Informedia@TRECVID 2016 MED and AVS

Junwei Liang, Poyao Huang, Lu Jiang, Zhenzhong Lan,
Jia Chen and Alexander Hauptmann
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We report on our system used in the TRECVID 2016 Multimedia Event Detection (MED) and Ad-hoc Video Search (AVS) tasks. On the MED task, the CMU team submitted runs in 000Ex, 010Ex and 100Ex settings for the Pre-specified Events. On the AVS task, the CMU team submitted runs for fully-automatic system with no annotation condition.

1 MED System

There are 3 tasks in MED this year: 000Ex, 010Ex and 100Ex. We designed two different systems for 000Ex task and 010Ex/100Ex task. The 000Ex task system is very different from the other since it does not utilize any training data. In the following section, we will describe our systems separately.

1.1 010Ex/100Ex System

The MED system for 010Ex and 100Ex consists of feature representations, model training, model transformation and fusion. System components are shown in Figure 1. We extract a variety of low-level and high-level features for feature representations. Here we describe these components in detail.

Low-level Features We extract the standard MFCC features and encode them using bag-of-word representations. Improved Dense Trajectories are extracted using the standard library [13]. Two deep convolutional neural network (DCNN) features, the VGG net [9] and the Residual net [3], are extracted. We use the 19-layer version of the VGG net and 152-layer of the Residual net. We concatenate the features of the fully connected layers (fc6 and fc7) of VGG net and the features of the pool5 and prob layers of the Residual net. We first extract DCNN features from the keyframes of the videos then use average-pooling to get video-level features. We utilize explicit feature mapping [12] (order 3 with intersection kernel) to expand the DCNN features into higher dimension to avoid using kernel classifiers for speeding up.

High-level Features The SIN [1], YFCC [10] and Sports1M [6] high-level features are extracted using last year's improved dense trajectories based system. These three semantic features are concatenated to form as IDT-Semantic feature. We train new semantic features using dataset from FCVID [5] and Activity Net [2]. A total of over 110k videos of 439 classes from these two dataset are utilized to train models using self-paced curriculum learning [7; 4]. The low-level feature we use is 19-layer VGG net with explicit feature mapping as before. The semantic features are concatenated into a 439-dimensional features as VGG19-Semantic for the final runs.

Model Training After feature representations are ready for each video, we train one-versus-all SVM models with self-paced curriculum learning [7; 4]. Noted that the miss videos from the event kit are used in the training and they are considered as "hard" samples that the model will learn later in the process. The hyper-parameters of the models are set via k-fold cross-validation (5-fold for 010Ex, 10-fold for 100Ex).

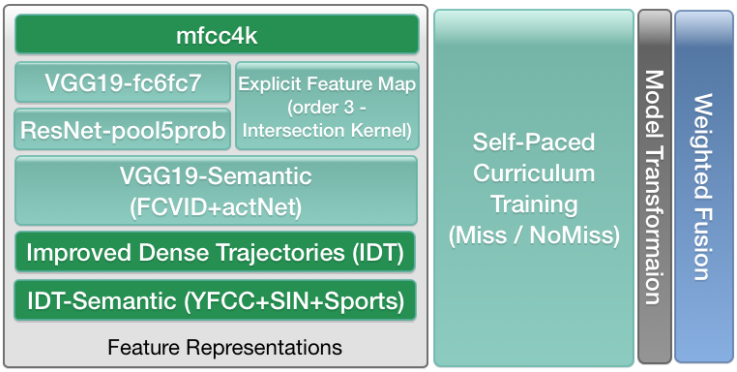


Figure 1: MED 010Ex/100Ex system components. The light green components or features are new compared to last year's system.

Table 1: Features used in our 010Ex/100Ex system. BoW: bag-of-words representation. DCNN: Deep Convolutional Neural Network

	Low-level Features	High-level Features
Feature Representations	MFCC (BoW) Improved Dense Trajectories [13] DCNN - VGG Net [9] DCNN - Residual Net [3]	Semantic Indexing Concepts (SIN) [1] YFCC [10] Sports1M [6] FCVID [5] Activity Net [2]

Model Transformation and Weighted Fusion We transform the SVM models into linear primal form in order to speed up the event search phrase. A weighted late fusion of all the output of feature models is used to produce the final results. The weights of the late fusion are learned through k-fold cross-validation (5-fold for 010Ex, 10-fold for 100Ex).

1.1.1 Submitted Runs

p-crossVal This is the primary run that utilizes all features and train one-versus-all SVM models using self-paced curriculum algorithm with all positive, miss and background videos. The training algorithm is an iterative process and the best model is selected based on cross-validation.

c-selfVal In this run, all features are utilized and trained with standard SVM algorithm using all the videos in the event kits.

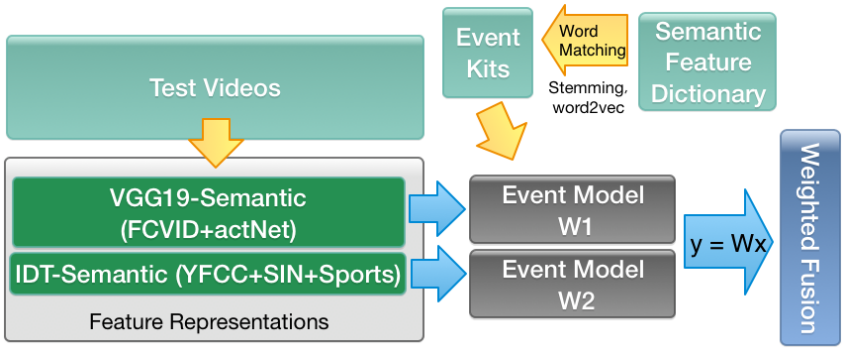


Figure 2: MED 000Ex system components.

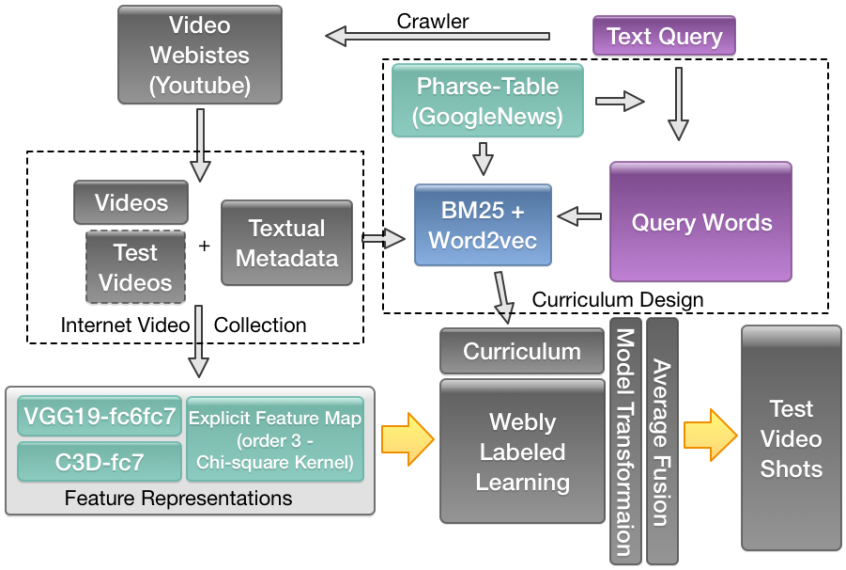


Figure 3: AVS system components.

1.2 000Ex System

The 000Ex system takes the textual event kit as the input, and outputs a ranked list of relevant videos. This year's system, as shown in Figure 2, is much simpler than last year's. During semantic query generation, we use stemming and word2vec to match the event kits' word to the semantic feature vocabulary, and form a linear regression model for each semantic feature. The semantic features are extracted as in the 010Ex/100Ex system. During event search phrase, the final result is a weighted fusion of the two semantic feature models' output. The weights of the fusion is empirically set based on the observation that the VGG19-Semantic features are better than the IDT-Semantic so we set the weights to be 6 to 4.

2 AVS System

In this year's new Ad-hoc Video Search task, we design a fully-automatic webly-label learning system that requires no annotation to perform a user search on the test set. Detailed algorithm for curriculum design and model training can be referred to our webly-labeled learning paper [7].

2.1 System Description

Our system consists of video collection, feature extraction, curriculum design, model training and query search as shown in Figure 3.

Video Collection Since our system requires no manual annotation for ad-hoc queries, it automatically collects Internet videos based on the textual queries for training query models. Given a user query, our system first refines the queries (currently we only strip out the "find shots of" prefix of the official queries) suitable for the video crawler to search for relevant videos on popular video hosting sites like Youtube using their search engine API. Then the system downloads these videos along with their user-generated textual metadata (including titles, descriptions, comments, etc.) into our Internet Video Collection. The test videos (IACC.3) can also be included in this collection since they too have metadata. However, we didn't use that in our submission due to the quality being too low (very few meaningful metadata in the IACC.3 data).

Feature Extraction We extract keyframe-level deep convolutional neural network - VGG-19 net [9] features including fc6/fc7 layers and the fc7 layer features of the C3D [11] net and then form video-level representations by average-pooling. Explicit feature mapping [12] (order 3 with chi-square

kernel) is used to expand the features into higher dimension to avoid using kernel classifiers for speeding up.

Curriculum Design In curriculum design phrase, our system tries to rank the training videos by their relevance to the query from the Internet Video Collection based on the prior knowledge extracted from their textual metadata. Specifically, we consider each video's metadata as a document and utilize word2vec [8] and BM25 algorithm to retrieve the relevant videos. We use a phrase table extracted from GoogleNews corpus for word tokenization.

Model Training In model training phrase, we utilize webly-labeled learning algorithm [7] to learn one-versus-all query model, where the model is refined iteratively from easy to hard samples. The best model is selected based on empirically setting the selection threshold to 0.5 (It means that we will select the model trained with half of the total collection retrieved during the curriculum design phrase). The final model is transformed to primal form to speed up query search.

Query Search Finally, after query models are trained, we apply them to the test video shots that are longer than 3 seconds. Average late fusion is used for the final results.

2.2 Submitted Runs

INF_CMU_c3d+vgg This run utilizes the full AVS system and the final result is computed with average fusion of the output of VGG net and C3D net models.

INF_CMU_vgg This run uses the full AVS system but with only VGG net features.

INF_CMU_vgg_batchTrain This run utilizes the videos retrieved during curriculum design phrase as positive samples and trains a standard one-versus-all SVM model using VGG net features.

INF_CMU_semantic This run uses our MED 000Ex pipeline, where we match the text query to our semantic feature vocabulary to form a linear regression model of the semantic features extracted from the test video shots. The semantic features include $1433 + 439$ concept detectors trained from the YFCC [10], SIN [1], Sports1M [6], FCVID [5] and Activity Net [2] datasets.

References

[1] G. Awad, C. G. M. Snoek, A. F. Smeaton, and G. Quénot. Trecvid semantic indexing of video: A 6-year retrospective. *ITE Transactions on Media Technology and Applications*, 4(3):187–208, 2016. Invited paper.

[2] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 2, page 6, 2015.

[5] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[7] J. Liang, L. Jiang, D. Meng, and A. Hauptmann. Learning to detect concepts from webly-labeled video data. 2016.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767*, 2014.

[12] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):480–492, 2012.

[13] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

INF@TRECVID 2016: Surveillance Event Detection

Jia Chen[1], Jiande Sun[2],Yang Chen[3], Alexandar Hauptmann[1]
[1]Carnegie Mellon University
[2]Shandong University
[3]Zhejiang University

1 Introduction

We develop a mixed strategy to tackle different event types in the surveillance event detection. As Embrace, Pointing and Cell2Ear events have strong static visual cue, i.e. key pose, we propose a object detection approach. For other event types, we use the previous year's solution which predict event based on video shots.

2 Methodology

2.1 Key Pose detection for Embrace, Pointing and Cell2Ear

First, we sample one frame per second for Embrace, Pointing and Cell2Ear events. On this frames, we manually labeled the bounding box for the corresponding people involved in the event. Altogether we get $1,853$ bounding boxes for Embrace event, $2,518$ bounding boxes for Pointing event and $1,391$ bounding boxes for Cell2Ear event. We treat key pose as a special kind of object and reduce key pose detection problem to object detection problem. We use Faster-RCNN[1] to learn the key pose detection for Embrace and Pointing.

To discriminate Embrace, Pointing and Cell2Ear poses from other poses, we add an additional class called other pose as hard negatives. The label of this class is automatically generated from the pre-trained person detector. To be specific, we use the person class detector from Faster-RCNN trained on MSCOCO. We threhold the person class output by score 0.8. As shown in Figure 1, Faster-RCNN produces reasonable outputs on all camera scenes in SED.

In the test stage, we predict pose on images sampled per 10 frames. We set the threshold for the score at 0.1. Finally we apply average pooling on striding windows of width 50 frames and stride 50 frames.

2.2 Shot Classification for Other Events

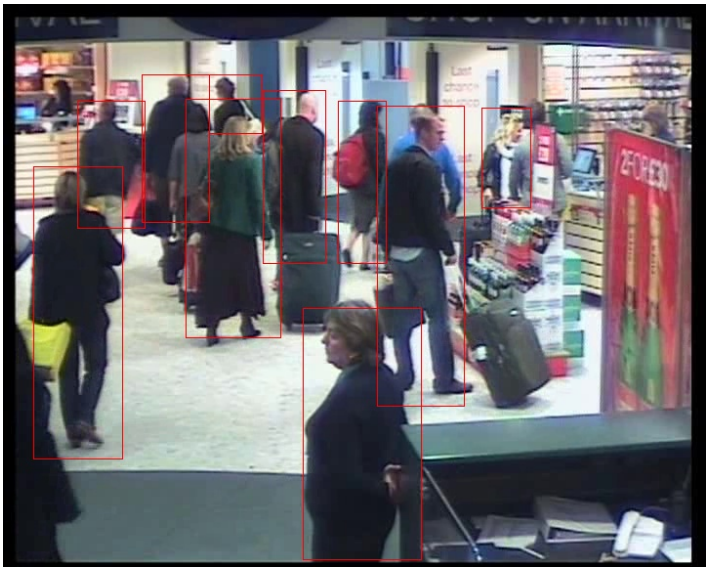
We extract non-overlapping 25-frame shots from videos. We extract Dense Trajectory Feature[2] and train a multi-class SVM for other event detection following our last year submission.

3 Experiment

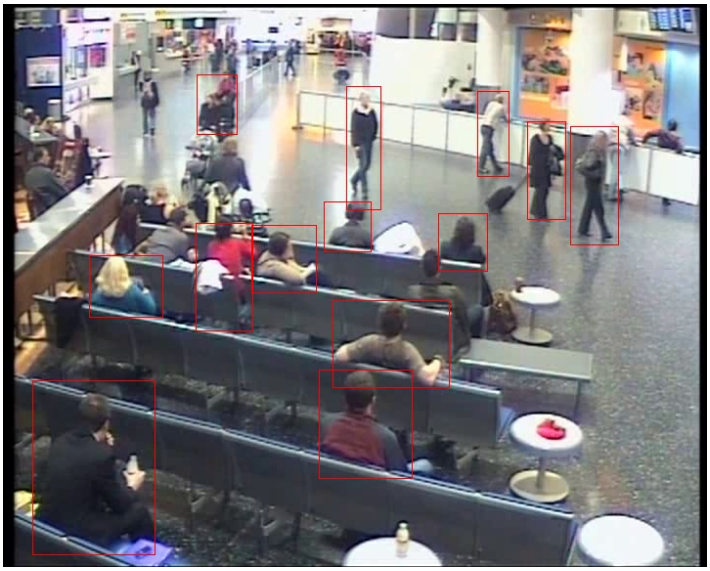
To train Faster-RCNN, we finetune it from a VGG16-based Faster-RCNN pretrained on MSCOCO. We report the average precision (AP) of Embrace, Pointing and Cell2Ear pose detection. We use AP as it is comparable to other object detector performance on PASCAL and MSCOCO. The test dataset is a subset of Eev08-1 by sampling images without Embrace, Pointing and Cell2Ear event to keep its ratio to images with Embrace, Pointing and Cell2Ear at $6:1$. Here we note that on the real video test dataset the negative to positive ratio is around $921:1$, much larger than the our sampled image test set. Fase positive is the major issue for the relatively low performance of current algorithm.

As shown in table 1, the performance of Embrace, Pointing and Cell2Ear is much lower than the object detector's performance reported on PASCAL and MSCOCO dataset. It indicates that Embrace, Pointing and Cell2Ear pose detection is much harder than general object detection. Considering that embrace and pointing pose is more find-grained than person detection, it is natural that directly applying Faster-RCNN doesn't achieve good performance.

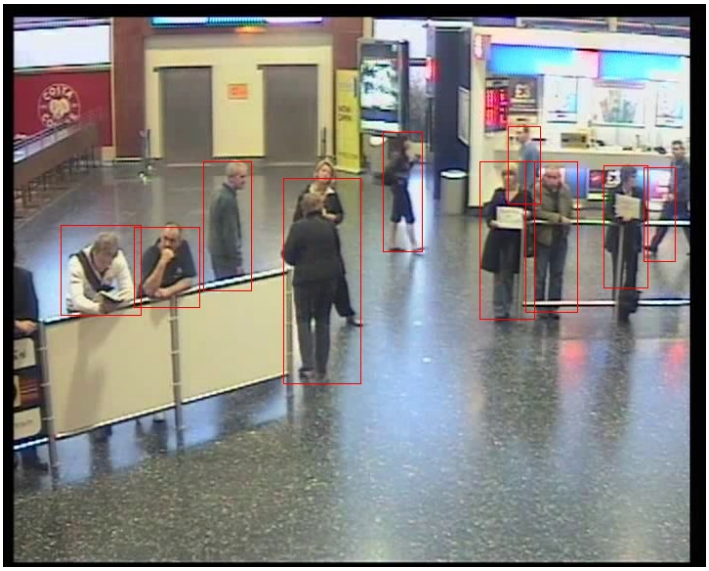
In addition to evalute the performance on image test set, we also evaluate the result on real video test set Eev08. As shown in table 2, we see that our algorithm achieves promising results for Embrace event on actual DCR. We further decompose the performance by actual RFA and actualPMiss to study the cause behind low performance of Pointing and Cell2Ear. We find that the major issue with



(a) CAM1



(b) CAM2



(c) CAM3

(d) CAM5

Figure 1: Person Detection on SED

Cell2Ear is that the model misses many positive instance as actuallPMiss is very high and it only detects 12 true positives. As for Pointing event, the model doesn't perform well on either actualRFA or actual PMiss, leading to the bad overall result.

Table 1: average precision of pose detection

Embrace	0.425
Pointing	0.263
Cell2Ear	0.024

Table 2: Evaluation on Eev08 by official metrics

	actualDCR	minDCR	actualRFA	actualPMiss	♯CorDet
Embrace	0.7335	0.7006	40.93	0.529	139
Pointing	0.9648	0.9550	22.33	0.853	254
Cell2Ear	0.9901	0.9308	5.57	0.962	12

References

[1] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[2] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011.

INF@TRECVID 2016 Video Hyperlinking

Xuanchong Li, Alexander Hauptmann
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213
{xcli,alex}@cs.cmu.edu

Abstract

We report our video hyperlinking system which utilizes mutlimodal information. The major componet of our system is the language-aided multimodal retrieval (LAMAR) framework which introduces the language as the intermediate video representation. Our submissions are combinations of multimodal features and retrieval methods under the framework of LAMAR. The results on development set and test set shows that LAMAR significantly outperforms the unimodal method.

1. Introduction

Most content-based video retrieval (CBVR) tasks focus on the similarity search given one event description. For example, the multimedia event detection (MED) task at TRECVID is aimed to retrieve videos that contains the described events; the surveillance event detection (SED) task at TRECVID is targeted to detect observations of events from surveillance video stream. In both of MED and SED, each query concentrates on just one type of concept or event. However, the video, as a very rich information source, usually contains numerous events or concepts. For example, the street-view video stream might include information from the simple concept such as people, cars, dogs to the compound event such as busking, traffic accident and so on.

On the other hand, CBVR benefits from the complementary information from diverse modalities of videos. However, recent benchmarks of VH at MediaEval and TRECVID show that the unimodal method on the speech channel is still dominating the VH systems. Many efforts of adding more modalities fail to improve the performance. In addition to the multimodal retrieval, crossmodal retrieval is another desired feature. For instance, a concept presented in the audio channel should be used to search in the visual channel.

The current limitations of VH research drive the inspiration of language-aided multimodal retrieval (LAMAR). As the name suggests, LAMAR employs the natural language, rather than a global numerical vector, as the representation of video contents. For instance, given a street-view video clip, a traditional CBVR system might represent it with a 1024-dimensional vector. Instead, LAMAR might represent it with a set of natural language description. E.g., "A man with white shirt is making a phone call", "A Scottie dog is walking", "A red phone booth", "A textual word 'Starbucks'", "A voice saying 'how are you doing?'", etc.. This natural language representation layer provides multiple benefits including

- *Offering a rich and complete distributed representation space*: the universal representation capability of natural language offers a well-suited representation space. Besides, the distributed nature of language keeps the local concepts/events individually rather than squeeze them into a global representation. This property accommodates LAMAR well in the VH setting.

- *Mutlimodal/crossmodal retrieval in a common language space*: since the information from every modality is projected into the language space, searching in the language space spontaneously renders the multimodal/crossmodal retrieval.

- *Transfer learning from the progress of textual retrieval*: natural language processing and textual retrieval have achieved significant progress. LAMAR can easily transfer these advances into the multimodal domain. For example, a semantic word embedding (e.g. word2vec) can help group the similar concepts together.

- *Naturally interpretable results*: LAMAR can easily trace back the relevance ranking to the intermediate language representation layer. This quality leads to a more interpretable system.

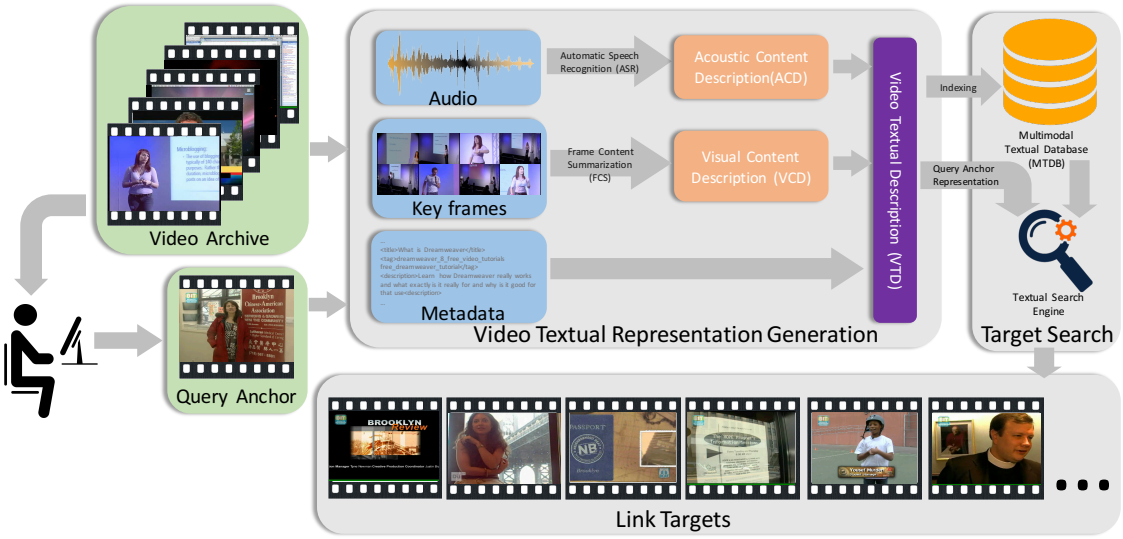


Figure 1. The architecture of LAMAR

2. Methods

The general approach of VH includes two steps: *anchor representation* and *target search* [13]. LAMAR follows the same procedure. Figure 1 shows the overall architecture of LAMAR. The *anchor representation* step corresponds to video textual representation (VTD) generation, during which we mapped the audio and visual information into natural language space separately. For the audio part, we use the *acoustic speech recognition (ASR)* to extract speech script from the soundtrack. For the visual part, we apply *frame content summarization (FCS)* on every keyframe with various visual-to-text methods such as the image concept detection[9], dense image captioning[6], and natural scene OCR[3][5]. In addition to the content-based information, we also incorporate the metadata such as title, tags and user-generated description into the VTD. The videos are converted to The *target search* step in LAMAR is a *textual search engine*. We integrated various methods in *textual search engine* from vector space models (e.g. TF-IDF[7]) to neural network based models (e.g. word2vec embedding[12]).

In the following subsections, we present the details in frame content summarization and textual search engine.

2.1. Frame Content Summarization(FCS)

Frame Content Summarization is for generating the visual content description for each keyframe. We combine three complementary methods: image concept detection, dense image caption, and natural scene OCR.

- *Image Concept Detection*: it comes with the dataset package. It provides the top-5 concepts and scores for each keyframe. The concept is from AlexNet trained on ImageNet dataset[9]. We use the concept words as the representation.

- *Dense Image Caption*: while the *Image Concept Detection* assumes the image contains a single concept, the video frame can contain much richer information. We employed the dense caption[6]. It generates descriptive caption from many local areas from a single frame. With a threshold, we extract the captions with a high score as the representation.

- *Natural Scene OCR*: besides the object concept/caption, there are many textual characters in the video. To extract this, we use an R-CNN style model to do natural scene OCR. Compared to the traditional OCR, which performs on a constrained scanned documents images, natural scene OCR is much more robust to the background noise and distortion. It first extracts some text proposals to localize the text areas[3]. Then a CNN trained on the dictionary is applied to recognize the word on those areas[5]. We take the high score words recognized as the representation.

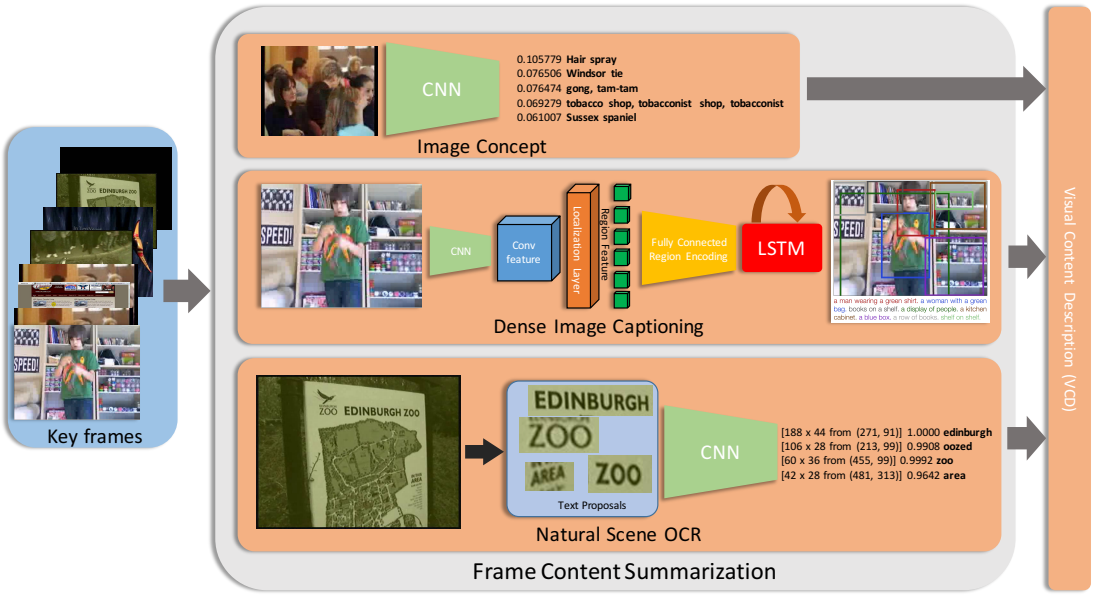


Figure 2. The Frame Content Summarization in LAMAR

We consider the FCS as an extensible module in LAMAR. Under this framework, many recent research efforts on bridging video/image and text, such as image captioning[8], video captioning[16], visual storytelling [4], and so on, can easily fit in the FCS to build a better VH system. TRECVID 2016's pilot task, video to text (VTT) [1], can potentially benefit the research of video hyperlinking.

2.2. Target Search

After getting the video textual description, we treat each video as a text document. The target search is a textual search engine. Regarding the representation of the text documents, we categorize our method into two groups, (a) vector space models and (b) word embedding based methods. For all the document representation methods, we use the cosine similarity as the relevance score.

2.3. Vector Space Models

The vector space models are the models represent the document as a vector of terms. It means each dimension of the vector corresponds to a term in the vocabulary. The value of on each dimension depends on various weighting method. The most famous one is the TF-IDF weights[7]. We explore various weighting models including BB2, BM25, DFR-BM25, DLH, DLH13, DPH, DFRee, Hiemstra-LM, IFB2, In-expB2, In-expC2, InL2, LemurTF-IDF, LGD, PL2, and TF-IDF[1].

2.4. Word Embedding Based Models

This document representation is built on the single word representation. The word2vec embedding[12] shows unprecedented performance on many tasks. For each word in the documents, it trained a shallow neural network to predict nearby words. Despite its simplicity, it captures amazingly complex relationship among the terms such as $v(king) - v(man) \approx v(queen) - v(woman)$. This property can help cluster similar concepts together. After mapping the words into embedding space, we take the average pooling to get the document representation.

3. Experiments

We perform the experiment on the development anchors with various feature combinations and method choices. Here are some details about our experiments.

3.1. Dataset

The TRECVID 2016 Video Hyperlinking dataset consists of 14,838 videos for a total of 3,288 hours from blip.tv. The data is accompanied by metadata, two kinds of ASR

[1]See `http://terrier.org/docs/v4.1/javadoc/org/terrier/matching/models/package-summary.html` for details of each model

transcripts (generated by LIMSI [10], LIUM [15] respectively), image concept detection from AlexNet[9].

The development set contains 28 query anchors. For each of them, a set of ground-truth anchors is provided. The test set contains 94 query anchors, which are for the final evaluation in the competition.

3.2. Experiment Setting

We use the fixed length segmentation of 50 seconds to convert videos into short video segments. For each segment, we run the FCS on the keyframes inside the segment. We use Terrier IR system[11] with default parameters for the vector space models. We use the word2vec model trained on Google News dataset[2].

4. Results

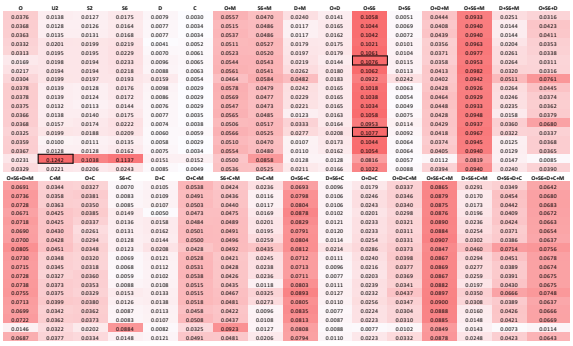


Figure 3. MAiSP on Development Set

Figure 4. mAP on Development Set

[2]Available at https://code.google.com/archive/p/word2vec/

Figure 5. P@10 on Development Set

Feature	Method	MAiSP	mAP	P@10
O+S6+C+M	Lemur TF-IDF	**0.136**	**0.243**	**0.367**
S6+M	DPH	0.095	0.135	0.318
U2	Word2Vec	0.076	0.087	0.216
O+S6	LGD	0.078	0.104	0.243
O+S6	DPH	0.082	0.111	0.251

Table 1. Performance on Test Set

We report the mAP, precision at top-10, and MAiSP[14] on both development set and test set. Figure 444 shows the performance numbers on with different feature combinations and methods. We use the following abbreviation: O=Natural Scene OCR; U2=LIUM12 ASR; S2=LIMSI12 ASR; S6=LIMSI16 ASR; D=Dense Caption; M=Meta Data. Table 1 shows our submission results. Note that previous best systems only use the ASR and metadata as the feature and simple TF-IDF as retrieval method. Introducing more modality by fusing retrieval results from different feature does not improve over the unimodal system[2]. In our experiment on both development set and test set, the multimodal methods outperform unimodal methods by a significant margin. It shows that LAMAR can utilize the multimodal sources effectively.

5. Conclusions

In the paper, we present our language-aided multimodal retrieval (LAMAR) framework for video hyperlinking. LAMAR introduces the natural language as the intermediate representation layer for video. LAMAR cure the lameness of current VH system from two aspects. First, it provides a distributed representation for the rich concepts/events in the video. With such language representation, multiple concepts/events in a single video segment can be maintained throughout the retrieval process. Second, LAMAR can automatically perform multimodal/crossmodal retrieval by searching in the same language space. The experiment results on development set and test set shows that LAMAR significantly outperforms

unimodal methods.

References

[1] G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F. Smeaton, G. Qunot, M. Eskevich, R. Aly, and R. Ordelman. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings of TRECVID 2016*. NIST, USA, 2016.

[2] Z. Cheng, X. Li, J. Shen, and A. G. Hauptmann. Which information sources are more effective and reliable in video search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1069–1072, New York, NY, USA, 2016. ACM.

[3] L. Gomez-Bigorda and D. Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *CoRR*, abs/1604.02619, 2016.

[4] T. K. Huang, F. Ferraro, N. Mostafazadeh, I. Misra, A. Agrawal, J. Devlin, R. B. Girshick, X. He, P. Kohli, D. Batra, C. L. Zitnick, D. Parikh, L. Vanderwende, M. Galley, and M. Mitchell. Visual storytelling. *CoRR*, abs/1604.03968, 2016.

[5] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision*, 116(1):1–20, Jan. 2016.

[6] J. Johnson, A. Karpathy, and F. Li. Densecap: Fully convolutional localization networks for dense captioning. *CoRR*, abs/1511.07571, 2015.

[7] K. S. JONES. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[8] A. Karpathy and L. Feifei. Deep visual-semantic alignments for generating image descriptions. 2015.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[10] L. Lamel. Multilingual speech processing activities in quaero: Application to multimedia search in unstructured data. In *Baltic HLT*, pages 1–8, 2012.

[11] C. Macdonald, R. McCreadie, R. L. Santos, and I. Ounis. From puppy to maturity: Experiences in developing terrier. *Proc. of OSIR at SIGIR*, pages 60–63, 2012.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119, 2013.

[13] R. J. Ordelman, M. Eskevich, R. Aly, B. Huet, and G. Jones. Defining and evaluating video hyperlinking for navigating multimedia archives. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 727–732. International World Wide Web Conferences Steering Committee, 2015.

[14] D. N. Racca and G. J. Jones. Evaluating search and hyperlinking: an example of the design, test, refine cycle for metric development. In *Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany*, 2015.

[15] A. Rousseau, P. Deléglise, and Y. Estève. Enhancing the ted-lium corpus with selected data for language modeling and more ted talks. In *LREC*, pages 3935–3939, 2014.

[16] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. *CoRR*, abs/1505.00487, 2015.

INF@TRECVID 2016: Video Caption Pilot Task

Jia Chen, Pingbo Pan, Poyao Huang, Hehe Fan, Jiande Sun, Yi Yang, Qin Jin, Alexander Hauptmann

1 Introduction

As the video caption pilot task provides no training captions for videos, we treat it as an opportunity to test the generalization ability of the caption models. That is, we don't tune the models to consider "who", "what", "where" and "when" facets explicitly. To be specific, we train four caption models on three public datasets and most of them achieve state-of-the-art result on these public datasets. It shows that there are much space left to improve current state-of-the-art models by considering these four facets explicitly.

2 Methodology

Our submission includes four models trained on three public datasets: MSCOCO[3], MSVD[1] and MSR-VTT[8]. MSCOCO is a popular image caption dataset used to benchmark image caption models. MSVD is a popular video caption dataset used to benchmark video caption models. MSR-VTT is a recently released video caption dataset that is more diverse on video categories than MSVD dataset.

2.1 Image Caption Model

We use the multimodal image caption model[4] on MSCOCO. The structure is illustrate in Figure 1. In test, we extract the middle frame from the video and apply image caption model on it. The features used in this model include VggNet[6].

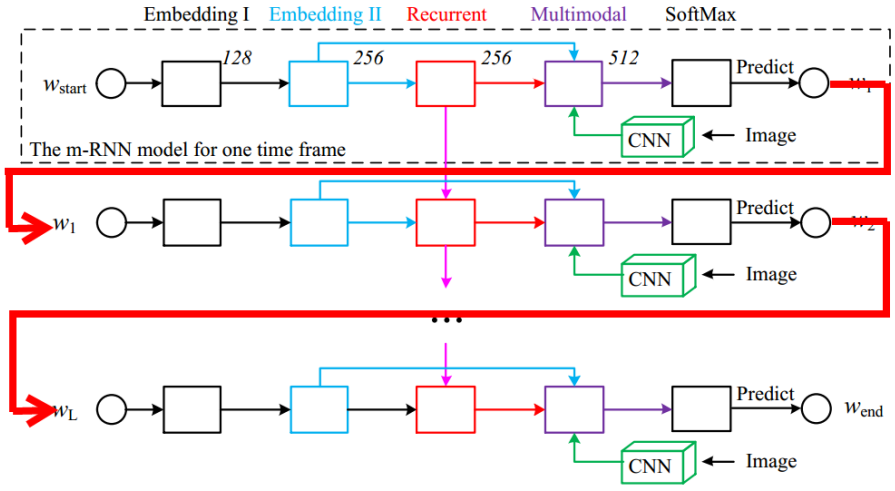


Figure 1: Multimodal image caption model

2.2 Hierarchical Recurrent Neural Encoder (HRNE) for Video Caption

Different from image, video contains not only spatial information but also temporal information. We use Hierarchical Recurrent Neural Encoder (HRNE)[5] to incorporate temporal structure in the encoder. The structure is illustrated in Figure 2. We add attention units in three different positions: between visual input and the LSTM filter, between the output of the filter and the second LSTM layer, between the output of our HRNE and the description decoder. Please refer to work[5] for more detailed description. The features used in this model include ResNet[2].

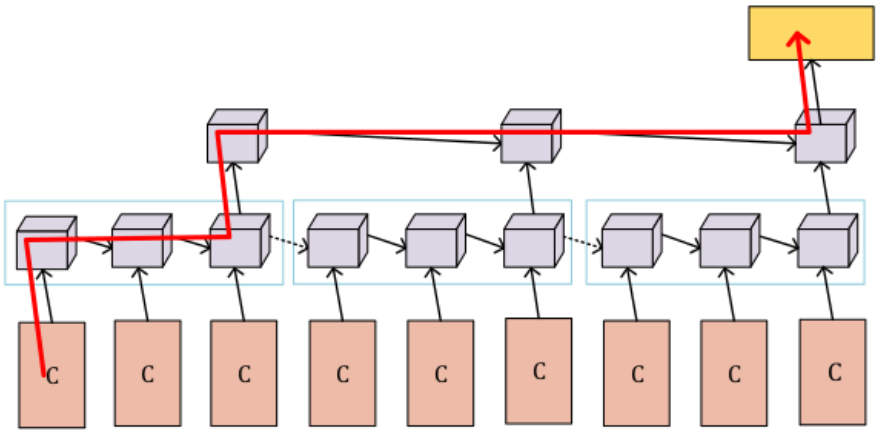


Figure 2: Hierarchical recurrent nueral encoder

2.3 Category-aware Multi-modality Fusion Encoder (CMFE) for Video Caption

To build a video caption model that words on various real-world videos, we could not neglect the multi-modality nature of videos. We design a category-aware multi-modality fusion encoder (CMFE) which does fusion adaptively for different categories. To be specific, we use gating function g_i of category vector s to switch feature fusion in different categories. The features used in this model include ResNet[2], C3D[7], MFCC[].

$$g_i = \sigma(\mathbf{U}^{(i)}s + \mathbf{B}^{(i)}) \tag{1}$$

$$h_0 = \sum_i g_i \circ tanh(\mathbf{W}^{(i)}f_i) \tag{2}$$

where f_i is feature i, s is the predicted category score vector. $\mathbf{U}^{(i)}, \mathbf{B}^{(i)}, \mathbf{W}^{(i)}$ are the parameters to learn.

3 Experiment

Our submited four runs include:

INF.coco image caption model trained on MSCOCO

INF.hrne HRNE trained on VTT

INF.jiac-msvd CMFE trained on MSVD

INF.vtt.iresnet CMFE trained on VTT

We first verify that the submitted runs achieve state-of-the-art performance on the public dataset MSVD. As show in table **??**, HRNE (MSVD) and CMFE (MSVD) both achieve state-of-the-art performance on MSVD if they are trained on MSVD. However, the performance of CMFE (VTT) trained on VTT is much worse than CMFE (MSVD). There are two factors that contribute to this phenomenon. First, the distribution of videos are different between MSVD and MSR-VTT. Second, the language style is different between MSVD and VTT.

Table 1: Performance on MSVD

method	Bleu1	Bleu2	Bleu3	Bleu4	Meteor	Cider
HRNE (MSVD)	81.1	68.6	57.8	46.7	33.9 -	
CMFE (MSVD)	80.4	68.6	58.4	47.5	34.1 78.2	
CMFE (VTT)	67.1	50.2	37.4	26.2	27.7	37.7

We compare the performance on the pilot task dataset in table 2. First the performance of CMFE (MSVD) on meteor drops significantly, 10%, from that on MSVD. The groundtruth on MSVD contains around 20 sentences for each video while the groundtruth on the pilot task contains only 2 sentences for each video. This will cause meteor evaluation to drop to some extent as there are fewer groundtruth sentence to match. But groundtruth sentence number is definitely not the only reason. The distribution of videos and language style could also contribute to the performance drop. As videos from VTT cover most categories of videos online,

video distribution is not likely to vary a lot between VTT and pilot task dataset. We will further verify this by manually labeling the category on pilot task dataset. Then the remaining factor, language style, is likely to contribute the most to the performance drop. In the groundtruth collecting of pilot task, annotators were asked to include and combine in one sentence, if appropriate and available, four facets of the video they are describing. This is likely to cause the language style to be very different from the public datasets that our models are trained on. Thus, we need to explicity model the four facets in the caption model to achieve better performance.

Table 2: Performance on Pilot task

method	Bleu	Meteor
MSCOCO	0.0073	0.1867
HRNE (VTT)	0.0140	0.2000
CMFE (MSVD)	0.0226	0.1974
CMFE (VTT)	0.0124	0.2083

References

[1] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2712–2719, 2013.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014.

[4] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014.

[5] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[7] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[8] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Acknowledgement

This material is based in part upon work supported by the National Science Foundation under Grant Number IIS- 1638429. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.