Informedia @ Trecvid 2017

Jia Chen[1], Junwei Liang[1], Jiang Liu[1,2], Shizhe Chen[3], Chenqiang Gao[2], Qin Jin[3], Alexander Hauptmann[1]

Carnegie Mellon University[1]
Chongqing University of Posts and Telecommunications[2]
Renmin University of China[3]

Informedia@TRECVID 2017
MED and AVS

Junwei Liang and Alexander Hauptmann
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We report on our system used in the TRECVID 2017 Multimedia Event Detection (MED) and Ad-hoc Video Search (AVS) tasks. On the MED task, the CMU team submitted runs in 010Ex settings for the Pre-specified and Ad-hoc Events. On the AVS task, the CMU team submitted runs for fully-automatic system with no annotation condition.

1 MED System

There are 1 task in MED this year: 010Ex. We use similar pipeline as last year but with new low-level features and semantic features.

1.1 010Ex System

The MED system for 010Ex consists of feature representations, model training, model transformation and fusion. 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 only one DCNN low-level feature this year. We use the Inception Resnet model [7] pre-trained on ImageNet. We first extract DCNN features from the keyframes of the videos then use average-pooling to get video-level features. pool5 layer output (2048 dimension) and the prob layer output (1000) are used and concatenated. We utilize explicit feature mapping [9] (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 [8] , Sports1M [4], UCF101, FCVID, places365 and Kinetics 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 our semantic feature with our Mixture-of-Residual-Expert model [2].

Model Training This year we didn't use any of the training data and treated the task as 000Ex.

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 1, 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.

1.2.1 Submitted Runs

p-ek0 This is the primary run that utilizes all features and uses the 000Ex system.

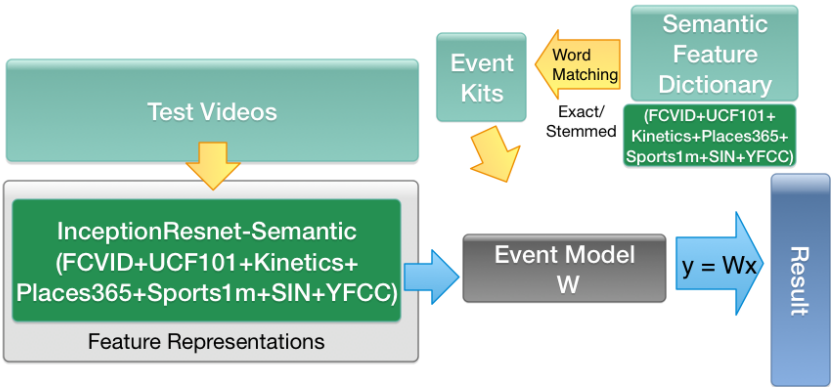

Figure 1: MED 000Ex system components.

Table 1: Features used in our 010Ex system. DCNN: Deep Convolutional Neural Network

	Low-level Features	High-level Features
Feature Representations	Inception Resnet [7]	Semantic Indexing Concepts (SIN) [1] YFCC [8] Sports1M [4] FCVID [3] Kinetics UCF101 Places365

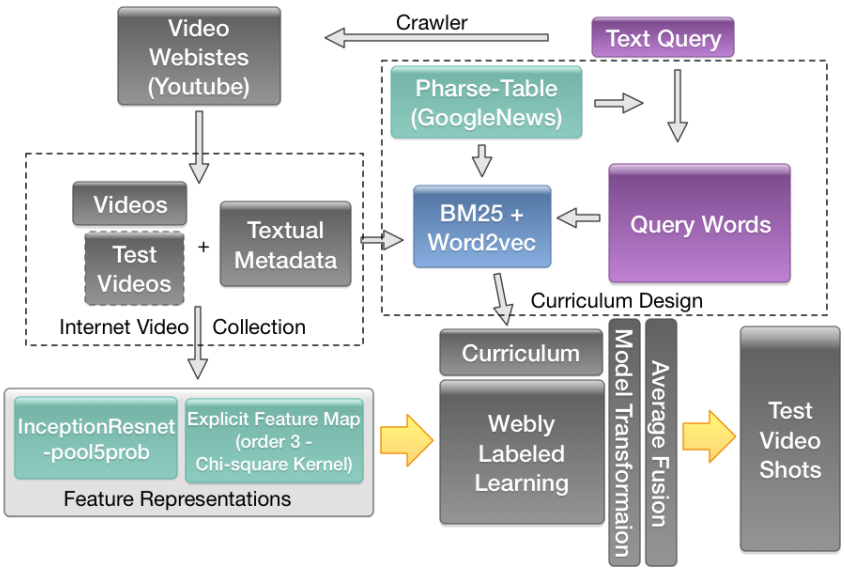

Figure 2: AVS system components.

2 AVS System

In this year's 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. The system is the same as last year. Detailed algorithm for curriculum design and model training can be referred to our webly-labeled learning paper [5].

2.1 System Description

Our system consists of video collection, feature extraction, curriculum design, model training and query search as shown in Figure 2.

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 use the Inception Resnet model [7] pre-trained on ImageNet. We first extract DCNN features from the keyframes of the videos then use average-pooling to get video-level features. pool5 layer output (2048 dimension) and the prob layer output (1000) are used and concatenated. Explicit feature mapping [9] (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 [6] 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 [5] 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 p (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_yt0.5 This run utilizes the full AVS system and the noise assumption is 50%.
INF_CMU_yt0.3 This run utilizes the full AVS system and the noise assumption is 70%.
INF_CMU_yt0.7 This run utilizes the full AVS system and the noise assumption is 30%.
INF_CMU_yt1.0 This run utilizes the full AVS system and the noise assumption is 0%.

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] P.-Y. Huang, Y. Yuan, Z. Lan, L. Jiang, and A. G. Hauptmann. Video representation learning and latent concept mining for large-scale multi-label video classification. *arXiv preprint arXiv:1707.01408*, 2017.

[3] 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.

[4] 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.

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

[6] 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.

[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017.

[8] 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.

[9] 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.

Informedia@TRECVID 2017 Surveillance Event Detection

Jia Chen[1], Jiang Liu[1, 2],Chenqiang Gao[2], Alexander Hauptmann[1]
[1]Carnegie Mellon University
[2]Chongqing University of Posts and Telecommunications

1 Introduction

Event detection is about spatial-temporal localizing the event and classifying the event. This shares some similarity with object detection, which is about localizing the object and classifying the object. Motivated by this similarity, we extends object detection pipeline to event detection pipeline on SED task, which is composed of four components: raw feature extractor, event proposal, event classification and event localization. In this year evaluation[4], our submission is generated by running the pipeline with baseline components.

2 Methodology

2.1 Pipeline overview

The pipeline is composed of four components:
raw feature extractor: it extracts features that are used in rest components.
event proposal: it filters out candidate tubes, which are temporal sequences of bounding boxes, for events. These tubes are called event proposal.
event classification: it classifies the event for each tube.
event localization: it refines the tube's temporal-spatial information.

Here's an analogy between event detection and object detection:
event proposal vs. object proposal: they are both about filter out as many irrelevant instances as possible when keeping recall at a high level so that more computation expensive components could be applied to these filtered instances.
event classification vs. object classification: they are both about classification and would benefit from many studies in this well separated task.
event localization vs. bounding box regression: they are both about precise localization based on the anchor position of the proposal.

The data flow between components are:
1. the output of raw feature extractor is used by all the rest components.
2. the output of event proposal is used by both event classification and event localization.
3. the event classification result could be used to help better event localization and vice versa. Thus these two components are interdependent.
This is summarized by the conceptual diagram of the pipeline in Fig 1.

2.2 Raw feature extractor

We extract three types of features: RGB image feature (VGG19), flow stream feature (GoogleNet) and RGB stream feature (C3D). As each frame contains several people and event only happens among some of them, we need to extract full convolutional feature in space to capture events at different locations in a frame. The details of time and space stride is summarized in Table 1.

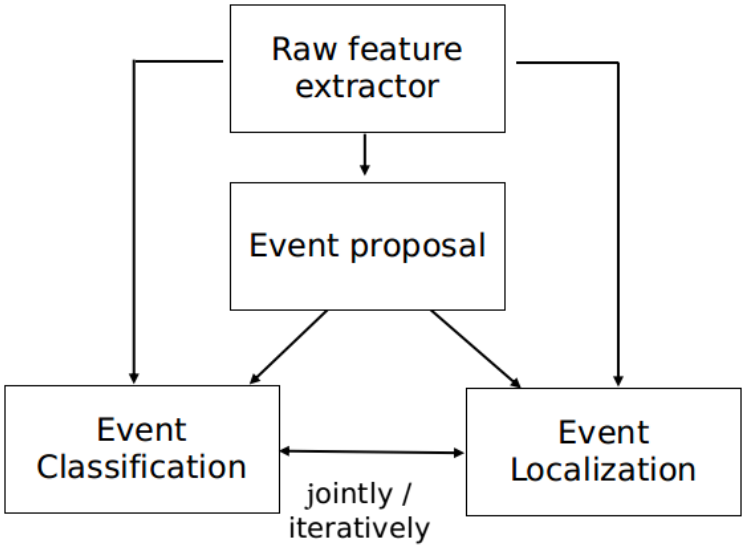

Figure 1: Pipeline Conceptual Diagram

Table 1: Details of extracted full convolutional features

feature	layer	time stride	space stride	spatial shape
c3d	conv5b	16	16	36×45
vgg19	pool5	5	32	18×23
flow stream	inception5b	6	32	18×23

2.3 Event proposal

As all events involve person, we start from person detection to get bounding box of person every 25 frames. This helps us filter out irrelevant regions in frames. However, a event proposal is a tube, a sequence of bounding boxes, rather than a bounding box. Thus, we use tracking algorithm to track the detected bounding box for a certain duration to get the tube corresponding to that bounding box. In each frame, we pad the tube to a square bounding box around its center trajectory. The padded tubes are event proposal candidates. Considering that tracking is not perfect, we do backward tracking to improve the recall of event proposals. We merge the event proposals from forward tracking and backward tracking by setting threshold of IoU to 0.5. As the duration of different events varies, we generate event proposals for two different time durations: 25 frames and 50 frames. An event groundtruth is considered as hit by an event proposal if the two tube intersect more than half of the tracking duration and all the IoU of bounding boxes in the intersected frames are larger than threshold. The details of recall under different theshold are summarized in Table 2.

2.4 Event classification

Given the event proposal tube, we retrieve feature points that lie in the tube from the full convolutional feature map. That is, we have a variant number of feature points for different tubes and and

Table 2: Recall of event proposals

duration	threshold	recall (%)
25	0.5	87.5
25	0.75	78.5
50	0.5	86.2
50	0.75	77.0

Table 3: Comparison of Pooling methods

method	Embrace	Pointing	Cell2Ear
mean	16.90%	5.50%	0.90%
SPP	10.00%	4.00%	2.40%
VLAD(32)	29.70%	9.40%	1.10%

Table 4: Comparison of Feature Fusion

feature	Cell2ear	Embrace	Pointing	PersonRuns	mAP
c3d	2.8	46.4	20.1	27.8	24.5
rgb stream	2.2	24.5	15.1	7.3	12.3
flow stream	2.9	40.9	35.7	59.7	34.8
c3d + rgb stream	3	44.2	20.6	27.6	23.9
rgb stream + flow stream	3.3	39.7	34.5	59.2	34.2
flow stream + c3d	3.9	50.6	34.2	58.8	36.9

need to encode them to get a fixed dimension feature representation for classification. We study three popular pooling methods, including mean pooling, spatial pyramid pooling (SPP)[5] and vlad pooling[6]. As the scale of the dataset is very large considering the amount of negative instances, we run a preliminary experiment on a subset of three events to select the best pooling method. We set the number of centers in VLAD to 32 and use 3 layer spatial pyramid pooling (4x4, 2x2, 1x1). To be specific, we set the negative to positive ratio on proposals to 10 in both training and testing. We measure the performance by average precision in each class. We use SVM as classifier. The result is shown in Table 3.

VLAD pooling significantly out perform mean and SPP on Embrace and Pointing events. SPP performs best on Cell2Ear althrough all three methods perform very pool on this event. One possible reason is that the classification clue of Cell2Ear event relies on the local arm movement which requires larger spatial resolution of feature map.

We further run a neural network version of VLAD called NetVLAD[3] but don't get any performance gain. There are two possible reasons. First, the number of positive proposal instances in SED dataset is limited, around hundreds, and we observe that the performance of NetVLAD on validation set saturates very early in the training stage, which indicates that there are not enough data for NetVLAD to reach its best performance. Second, we fix the feature and don't do end-to-end tuning due to time limit and therefore don't exploit the full potential power of NetVLAD. Thus, we use VLAD as our pooling method.

Then, we do feature fusion and selection. Again, we run fast preliminary experiment on a subset of four events and set the negative to positive ratio to 5. The evaluation metric is average precision of each event. As shown in Table 4, flow stream is the single best feature and rgb stream is the single worst feature. One possible reason is that the negative instances in SED task are person doing other things, which is very difficult to be distinguished using general rgb stream feature alone. The best feature combination is flow stream + c3d.

2.5 Event localization

We run a simple baseline of event localization: maximum suppression along time axis. Our final result on the standard SED evaluation metric is shown in Table 5 compared to the best team in this year. Note that we use only one model for all cameras and don't do any finetuning for each camera. Our run CMU flow-1 ranked 2nd all the four events we submit in this year.

2.6 Analysis of computation resources used in the pipeline

The video processing usually consumes many computation resources. In addition to the conceptual diagram, we also give a computation diagram to see whether the computation cost in this pipeline is reasonable and affordable. We'll first look at storage. Basically, we need to store on disk three types of data: original videos, extracted raw features and event proposals. Storage of original videos could

Table 5: Evaluation on SED17 using minDCR metric

run	Embrace	PersonRuns	Pointing	Cell2Eear
BUPT-MCPRL	0.5996	0.626	0.9308	NA
CMU p-c3d.flow_1	0.8351	0.7507	0.9962	1.0005
CMU flow_1	0.7631	0.6676	0.9755	0.9763

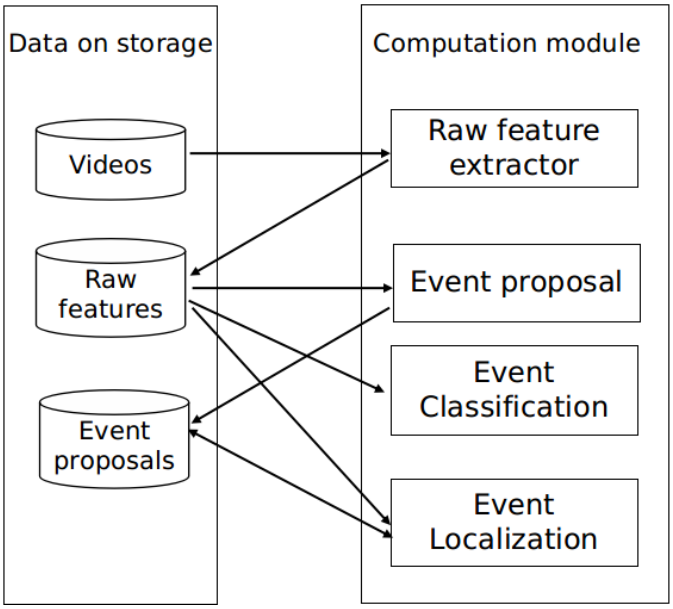

Figure 2: Pipline Computation Diagram

benefit from various video encoding algorithms. Raw features is the largest data and is the bottleneck for storage. After some optimization including using formats such as `npz` from numpy[1] and `tfrecord` from tensorflow[2] in storage, we could handle it when interacting with computation components in the pipeline. Event proposal is essentially sequences of bounding boxes and the storage could be neglected.

For the computation part, as the raw features are shared among all components, we avoid repeated computation of raw features. The computation cost of event proposal is also very high. Its computation cost comes from two sources. First, it need to look at all the content of video. Second it need to associate bounding boxes across time, which is similar to tracking in this sense. These two sources make the basic computation cost already very high without taking into account the complexity of algorithm itself. Thus simple and effective algorithm is preferred in the design of event proposal module. Both event classification and localization work on the event proposal and the scale of input is much smaller than that of original videos and raw features. Thus we could design some complex algorithms for these two components without any concern.

References

[1] Numpy. http://www.numpy.org/. Accessed: 2017-10-30.

[2] Tensorflow. https://www.tensorflow.org/. Accessed: 2017-10-30.

[3] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5297–5307, 2016.

[4] G. Awad, A. Butt, J. Fiscus, D. Joy, A. Delgado, M. Michel, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, M. Eskevich, R. Ordelman, G. J. F. Jones, and B. Huet. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *Proceedings of TRECVID 2017*. NIST, USA, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015.

[6] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3304–3311, 2010.

Informedia@TRECVID 2017
Video to Text Description

Jia Chen[1], Shizhe Chen[2], Qin Jin[2], Alexander Hauptmann[1]
[1]Carnegie Mellon University
[2]Renmin University of China

1 Introduction

For video caption, there are two kinds of approaches: retrieval and generation, and they correspond to the two subtasks in the VTT task of TRECIVD2017[2]. For the retrieval subtask, we focus on building models that have better discriminative ability as the model needs to distinguish between different captions give the video. For the generation subtask, we focus on testing the generability of the models as we observe that the performance on a fixed dataset has reached saturation to some extent.

2 Wide and Deep Models for Caption Retrieval

2.1 Wide Model

For caption retrieval, one important cue is whether a concept appears in the video. For m concepts, we use an m dimensional 0-1 feature to represent its appearance. As the number of total concepts is large while the number of concepts that appear in one video is small, this feature is very sparse. A simple linear model of dot product with the concepts has been widely used to ranking captions. The benefits of wide model is that we could reuse the result from concept detectors. We train a concept detector based on resnet feature for 1489 most frequently appeared concepts in MSCOCO.

2.2 Deep Model

Another branch of models in retrieval are multi-modal embedding models, which try to embed the caption and video into the same space. For video representation, we concatenate resnet and i3d features and do mean pooling across frames. For caption representation, we represent each word in the caption by word2vec. Note that this is not a fixed length representation and we will address this problem in the embedding model part.

For the embedding model, we design three models. For the first model, we calculate the embedding between each word and video and then apply mean pooling to get the final embedding score for the caption. We denote this model as mean pooling embedding (MPE). This model has two problems:
1. it treats each word separately. That is, even if we shuffle the words in the sentence, e.g. exchanging the subject and object, we still get the same final score.
2. each word, including stop word, contributes equally to the final embedding score.

In our second model, we address problem 1. we apply convolution operations with different kernel sizes on the words to learn the local structure of sentence. The kernel size correspond the to the length of local structure. To be specific, we use kernel sizes $1, 2, 3$. We apply mean pooling for convoluted result of each kernel size. Each one correspond to one different local structure length. We calculated the embedding result for each local structure length and added them together to get final embedding score. We denote this model as convolution embedding (CE).

Table 1: Comparison of retrieval models

method	feature	train set	testA	testB
CPE	mfcc+resnet200	MSRVTT	0.074	0.075
CE	mfcc+resnet200	MSRVTT	0.087	0.086
ACE	mfcc+resnet200	MSRVTT	0.094	0.089
ACE	resnet200+i3d	TGIF	0.112	0.107
Wide & Deep	resnet200+i3d	TGIF	0.115	0.109

In our third model, we improve upon CE to address problem 2. We use attention to do weighted linear combination of the embedding scores between each one in the convoluted result and video. The attention weight is a function of the convoluted result as we expect the model to learn to emphasize semantic local structure automatically. We denote this model as attention based convolution embedding (ACE).

2.3 Wide & Deep Model

We combine both deep model and wide model by adding the scores from both model and train them simultaneously.

2.4 Experiment

We train our model on TGIF and MSRVTT dataset and test them on TRECIVD 2016 caption retrieval set. The metric we used is mean inverted rank. As for deep models, we could see that the performance consistently improve (ACE $>$ CE $>$ CPE) by addressing more problems in the baseline CPE method. Training on TGIF dataset leads to better performance on TRECVID2016 test set. By combinining deep model with wide model, we see that the performance could be further improved.

3 Generalization Property of Caption Generation Models

For the generalization issue, we focus on two research questions:
1. Which one is more promising for better generalization on unseen datasets, high quality training dataset or more robust model?
2. Could we get more stable generalization ability by ensembling more different models?

To answer the first question, we could either fix the model architecture or the training datasets to study their influence by treating TRECVID2016[2] as unseen dataset. We compare two models: one is vanilla caption model with mean pooled video feature and another is temporal attention caption model[6]. We construct two training sets: one is the combination of MSRVTT[5] and MSVD[3]'s training part together and another is TGIF[4]'s training part. The results are summarized in Table 3. Note that we use the MSCOCO evaluation code[1] to calculate the caption metrics. We could see that the performance gain from dataset is much larger than the gain from the caption model. The number of video-caption pair in TGIF is on par with that in MSRVTT+MSVD. We note that in the collection process of TGIF, they have a detailed instruction such as "don't use any digits", "don't mention the name of movie and character", "don't mention invisible objects and actions", "don't make subjective judgements about TGIF". They disentangle advanced reasoning, e.g. invisible objects and actions, from the caption task. As current caption model is more focused on describing what it sees and hears from the video, this makes the dataset collected under these constraints more friendly for the model to learn. Thus the model learned on TGIF has a better generalization ability on TRECVID16 dataset.

To answer the second question, we get more models by varying the detailed settings such as tuning dropout rate and using different epochs in training. As shown in Table 3, we see that by ensembling more and more models from source domain datasets, the performance on the target domain dataset TRECVID16 improves consistently. It shows that ensembling also helps to improve the generalization ability.

Table 2: Comparison of changing model and change training sets

model	train dataset	BLEU4	Meteor	Cider
MP	MSRVTT+MSVD	5.04	12.13	30.25
ATT	MSRVTT+MSVD	5.59	12.38	31.96
MP	TGIF	8.05	14.67	37.00
ATT	TGIF	7.93	14.65	37.11

Table 3: Performance of ensembling

model	BLEU4	Meteor	Cider
best single model	8.05	14.67	37.00
ensemble 5 models	8.25	14.94	38.39
ensemble 6 models	8.25	15.04	38.66
ensemble 7 models	8.31	14.99	39.15
ensemble 8 models	8.46	15.04	40.79

References

[1] Mscoco evaluation code. https://github.com/tylin/coco-caption. Accessed: 2017-10-30.

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

[3] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR, June 2011.

[4] Y. Li, Y. Song, L. Cao, J. R. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. TGIF: A new dataset and benchmark on animated GIF description. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4641–4650, 2016.

[5] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5288–5296, 2016.

[6] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.