UTS_CAI submission at TRECVID 2018 Ad-hoc Video Search Task

Fengda Zhu, Xiaohan Wang, Linchao Zhu, Xuanyi Dong, Yi Yang
University of Technology Sydney
15 Broadway, Ultimo NSW 2007, Australia
{Fengda.Zhu,Xiaohan.Wang-3,Linchao.Zhu,Xuanyi.Dong}@student.uts.edu.au, Yi.Yang@uts.edu.au

Abstract

This work describes our approach used for the fully automatic Ad-hoc Video Search (AVS) task [7] for TRECVID[1] 2018. Our model is divided into two parts, visual model and language model. Our motivation is mapping video embedding and language embedding into same semantic space. We observe that by constructing triplets in the feature space we can take better advantage of large batches and hard examples. Our models are trained on MSR-VTT [12] and TGIF [5] dataset with different visual and language architectures. The final ensemble model achieves 6.7% mAP.

1. Introduction

We present our model for Ad-hoc Video Search (AVS) task, which is a task query video by natural language description in zero shot manner.

Our model is divided into two specific sub-models, video model takes video frames as input and outputs visual embedding and language model takes query sentence as input and outputs syntax embedding. This structure can be trained by minimizing the distance of visual embedding and syntax embedding. Thus, the problem of multi-modal query can be converted to searching the closet point from two sets.

There are lots of previous work done similar to this approach, Word2VisualVec[3] learns to predict a deep visual feature of textual input based on multi-scale sentences. VSE++[4] improve image-semantic embedding inference by hard example mining. These methods map inputs from different domains into same space. No matter what domain input come from, the more similar semantic information of inputs be, the more close distance their embedding be mapped.

In our work, we mainly focus on the optimization of visual and syntax embedding. The better embedding present semantic of video or query language, the higher performance our model get. The capability of different video and language encoders are experimented in this work.

To further improve the performance of our model, we use additional training data which comes from MSR-VTT and TGIF. We train model on each dataset individually and ensemble by late fusion[3] to achieve the best performance.

2. Proposed Framework

2.1. Video Clip Embedding

In order to obtain better representations of video clips, we utilize deep Convolutional Neural Networks to extract the semantic features of key frames and then aggregate them together. We sample five frames per second uniformly as the key frames of the video clips to capture the diversity and avoid redundancies. In consideration of the generalization of image recognition models, we use ResNeXt101[11] and PNAS[6] as our frame feature extractor, which both achieve promising performance on ImageNet Benchmark. Let $\phi(x_i;\theta_v)$ represent the CNN model, x_i is the ith input frame of video clip x and θ_v denote the model parameters. After that, we apply Average Pooling to these features and obtain final video clip representation:

$$\Phi(x;\theta_v) = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i;\theta_v)$$

Where n is the number of key frames in a video clip.

2.2. Joint Language Embedding

We investigate several language model to encode language information. Consider the structure of natural language, we use a joint method to generate language embedding. y is the input query sentence and y_i is a word in this sentence. First, we embedding each words in query sentence using Word2Vector (Each word is turned into a vector representation with a look-up table), converting each word to a vector in semantic space:

$$v_i = \Omega(y_i;\theta_w)$$

V is the collection of all the word embedding in a sentence, $v_i \in V$. After that, we apply sequential model to these features and obtain final language embedding,

$$\Psi(y;\theta_s) = Seq(V;\theta_{seq})$$

We use θ_w to represent the parameters of Word2Vec and θ_{seq} to represent the parameters of sequential model. θ_s stands for the parameter of the whole language model.

Word2Vec [8] is a wide used approach in language processing. It pretrained on large scale language dataset and best preserve the linear regularities among words. We regard it as a good warm up of word embedding and use it as weight initialization of word embedding.

We use two sequential model: RNN and TCN [2]. Embedding word vectors by LSTM [10] or GRU [9] is a common way in multi-modal problems [9] [13]. The resulting sequence of word embeddings is passed through a Recurrent Network. We use the last state of the RNN as the language embedding. We rather found it more effective to always run the recurrent units for the same number of iterations, including entries containing zero-padding.

In the bi-directional LSTM, each word corresponds to two hidden states, one for each direction. Thus, we concatenate its two hidden states to represent the semantic meaning of a word. Meanwhile, the last hidden states of the bi-directional LSTM are concatenated to be the global sentence vector.

2.3. Visual-Semantic Embedding

Ad-hoc Video Search Challenge is a cross-modal retrieval task. To address the problem, we employ two projection matrix W_v and W_s to map the visual feature and text feature to the joint space, respectively:

$$f_v(x;W_v,\theta_v) = W_v^T\Phi(x;\theta_v)$$

$$f_s(y;W_s,\theta_s) = W_s^T\Psi(y;\theta_s)$$

In our experiments, the CNN models are pretrained on ImageNet and parameters are fixed in visual-semantic embedding training procedure. Model parameters are represented as $\theta = \{W_v, W_s, \theta_s\}$.

We use consine distance to measure the similarity of two feature in the joint embedding space:

$$S(x,y) = \frac{f_v \cdot f_s}{\|f_v\|_2 \cdot \|f_s\|_2}$$

In order to pull the correct visual-semantic pairs closer and push the incorrect pairs away from each other, we use a triplet ranking loss to optimize our model:

$$L = \sum max[m - S(x,y) + S(x,\hat{y})], 0\} + \sum max\{[m - S(x,y) + S(\hat{x},y)], 0\} \quad (1)$$

The first term is taken the hardest negative text description given the query video clip, $\hat{y}$ means the closest incorrect description while y means the correct description. In the same way, the second term is taken the hardest negative video clip to compose the triplet. m serves as a margin parameter.

3. Experiments

We submitted 4 runs with different experimental setup. We present detail of these settings and the result of runs in the following sections.

3.1. Experimental Setup

There was no training data provided by NIST for the Ad-hoc Video Search (AVS). We adopt two video dataset, MSR-VTT [12] and TGIF [5], to train our model. We use TRECVID 2017 dataset as validation data. We slice each video of each dataset as frame with frame rate 5 per second.

MSR-VTT contains about 50 hours and 260K clip-sentence pairs in total, covering the most comprehensive categories and diverse visual content, and representing the largest dataset in terms of sentence and vocabulary. Tumblr GIF (TGIF), with 100K animated GIFs from Tumblr and 120K natural language descriptions obtained via crowdsourcing.

The TRECVID dataset [12] contains 4593 Internet Archive videos (600 total hours) using videos with durations between 6.5min and 9.5min. We have 30 query natural language sentence in total. In testing, for each sentence, our model returns the first 1000 videos according to semantic similarity.

3.2. Submitted Runs

We submitted four runs for Ad-hoc Video Search Task with fully automatic setting. Our four submitted runs evaluated under the metric of Mean extended inferred average precision (mean xinfAP). Table 1 shows the performance of our model on the TRECVID AVS test datasets. All these four runs are trained on MSR-VTT and TGIF.

In *RUN_1*, Language model uses Word2Vector to encode each word and LSTM to encode sentence . Visual model uses ResneXt-101 to encode visual input.

In *RUN_2*, we train 2 models with different visual model. ResNeXt-101 and PNAS are trained independently with same language structure. Then we ensemble these models together to get a great performance improvement.

In *RUN_3*, we train a single models which language model is consist with TCN, a more powerful language encoder, while visual model using ResNeXt-101.

In *RUN_4*, we train 2 models with different language model. LSTM, TCN are trained independently with same visual structure PNAS. Then we ensemble these models together as the same way as Run_2.

Table 1. The performances on test set

	RUN_1	RUN_2	RUN_3	RUN_4
infAP	0.031	0.067	0.058	0.067
iP5	0.227	0.293	0.280	0.327
iP100	0.157	0.231	0.203	0.235
iP1000	0.071	0.118	0.108	0.112

References

[1] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Qunot, J. Magalhaes, D. Semedo, and S. Blasi. Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *Proceedings of TRECVID 2018*. NIST, USA, 2018.

[2] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[3] J. Dong, X. Li, and C. G. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 2018.

[4] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. 2017.

[5] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650, 2016.

[6] C. Liu, B. Zoph, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. *arXiv preprint arXiv:1712.00559*, 2017.

[7] J. Lokoc, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad. On influential trends in interactive video retrieval: Video browser showdown 2015-2017. *IEEE Transactions on Multimedia*, 2018.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711*, 2017.

[10] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

[11] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[12] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016.

[13] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv preprint*, 2017.