NTU ROSE Lab at TRECVID 2018: Ad-hoc Video Search and Video to Text

Muhammet Bastan Xiangxi Shi Jiuxiang Gu Zhao Heng Chen Zhuo
Dennis Sng Alex Kot
ROSE Lab, Nanyang Technological University, Singapore

October 23, 2018

Abstract

This paper describes our participation in the ad-hoc video search and video to text tasks of TRECVID 2018. In ad-hoc video search, we adapted an image-based visual semantic embedding approach and trained our model on combined MS COCO and Flicker30k datasets. We extracted multiple keyframes from each shot and performed similarity search using the computed embeddings. In video to text, description generation task, we trained a video captioning model with multiple features using a reinforcement learning method on the combination of MSR-VTT and MSVD video captioning datasets. For the matching and ranking subtask, we trained two types of image-based ranking models on the MS COCO dataset.

1 Ad-hoc Video Search (AVS)

In the ad-hoc video search task, we are given 30 free text queries and required to return the top 1000 shots from the test set videos [1, 2]. The queries are given in Appendix A. The test set contains 4593 Internet Archive videos of 600 hours with 450K shots (publicly available on TRECVID website). Videos have a duration of 6.5 minutes to 9.5 minutes. The reference shot boundaries are publicly available. No annotated training data was provided specifically for the AVS task. We participated with a "fully automatic" (Type F) system trained on already available annotated datasets (Type D), MS COCO and Flicker30k [5, 6].

1.1 Visual Semantic Embedding

We adapted the visual semantic embedding, VSE/VSE++ [3, 4], for cross modal retrieval. Given a set of image-caption pairs, VSE++ learns a joint embedding space, which can be used for cross-modal retrieval, i.e., given text, retrieve images/videos or vice versa.

Figure 1 shows the VSE framework for learning a joint embedding between images and text, by maximizing the cross correlation between the image and text embeddings. First, image embedding is computed with a CNN, and text embedding is computed with an RNN using low dimensional word embeddings as input. Then, the similarity between the two embeddings are maximized by minimizing a triplet ranking loss. The similarity is computed as the inner product of the embeddings. During training, VSE++ samples hard negatives within each mini-batch and shows that using hard negatives improves the performance [4].

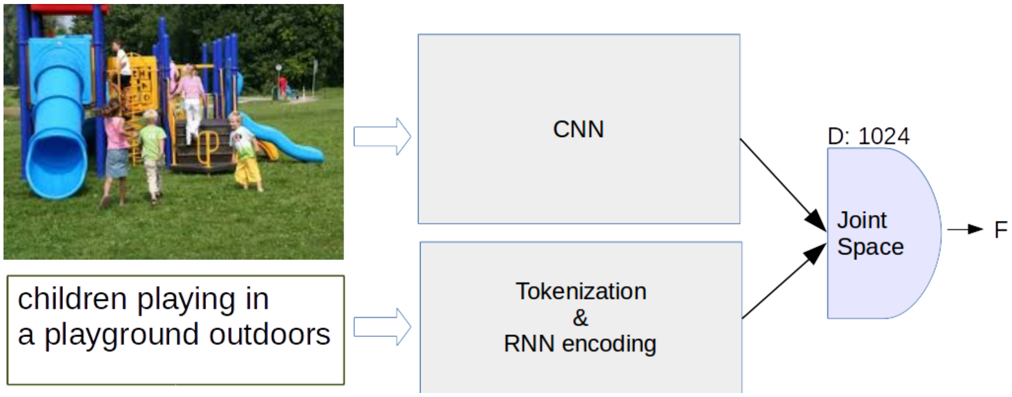
children playing in a playground outdoors
CNN
Tokenization & RNN encoding
D: 1024
Joint Space
F

Figure 1: Visual semantic embedding [3, 4] between images and text descriptions.

1.2 Training VSE++

We adapted the publicly available PyTorch implementation of VSE++ at `https://github.com/fartashf/vsepp` with a number of modifications.

- **Dataset:** We used the combination of MS COCO [5] and Flicker30k [6] images and captions datasets to increase the coverage and obtain a better model. Instead of sequentially training one one dataset and then fine tuning on the other, we trained on the combined dataset simultaneously. This is to avoid the catastrophic forgetting.
- **Vocabulary:** We constructed a single vocabulary as the combination of words in MS COCO and Flicker 30k.
- **Word and text embeddings:** We used a word embedding size of 1024, instead of the default 300. This is fed to a single layer GRU, which outputs a text embedding of size of 1024.
- **CNN:** We used ResNet152 with input image size of 224×224 and FC layer size of 1024, which is also the image/text embedding size.

- **Data augmentation:** We used the random resized crop (scale: 0.4-1.0, ratio: 0.9-1.1), random horizontal flip, color jitter (brightness: 0.2, contrast: 0.2) data augmentations from the *torchvision* library.

- **Training:** We used the training and validation set of MS COCO and training set of Flicker30k. We first froze the convolutional layers of the CNN and trained the FC layer along with the RNN, then fine tuned all layers of both networks. We used a mini-batch size of 300, and Adam optimizer with learning rates of 2×10^{-4} (15 epochs) and 10^{-6} (15 epochs) in first stage and fine tuning respectively. We saved the best model based on the highest validation accuracy (sum of R@k) on MS COCO and Flicker30k ($0.6 \times \sum R@k_{coco} + 0.4 \times \sum R@k_{f30k}$). We used this model to perform the queries on the TRECVID test set.

1.3 VSE++ on TRECVID Test Set

We trained the VSE++ model as described above. Using this model, we compute the keyframe embeddings of the shots and text queries and finally perform the queries and return top 1000 results for each query.

- **Keyframe extraction:** The shot boundaries, as well as keyframes of the test set were provided. Instead of using the provided single keyframes for each shot, we extracted multiple (up to 10) keyframes from each shot. We first sampled a frame at every 10 frames [7], then used DBSCAN algorithm [8, 9] to cluster the frames and obtain one or more keyframes for each shot. This is to have a better representation for each shot.

- **Keyframe and query embeddings:** Using the trained CNN model, we compute and save the keyframe embeddings to disk. We first resize the keyframe so that the smaller side is 230 pixels, then center crop 224×224 to feed into the CNN. Similarly, we compute the query embeddings using the trained RNN model and combined vocabulary.

- **Similarity search:** Once the keyframe and query embeddings are computed, we perform the similarity search, by computing the similarity between each query embedding and all keyframes using inner product and returning the most similar 1000 keyframes/shots. If the shot has multiple keyframes, the most similar keyframe is taken as the representative. We used exhaustive search and did not employ any approximate nearest neighbor search algorithm. On the average, query encoding took 0.0126 seconds, similarity search took 0.1358 seconds (total 0.1484 seconds) per query.

1.4 AVS Results and Discussion

We submitted a single run using the approach described above and achieved an inferred average precision **infAP** value of **0.082**, ranking 7^{th} among all the submissions (of type F&D), and 3^{rd} among the best submissions of each team, as shown in Figure 2 [2]. Overall, the infAP values of all the teams are quite low, the highest being 0.121.

The inferred average precision (infAP) values for each of the 30 test queries (Appendix A) is shown in Figure 3. In parallel with the top performing submissions, we performed well on some queries (e.g., 563, 565, 577, 584) and poorly on others (e.g., 567, 569, 576, 582, 588, 589, 590). This is due to (1) coverage of the training datasets and domain difference, (2) limited global image-based representation of the CNN, (3) lack of spatio-temporal video and audio modeling. Lack of large video datasets annotated at shot level is probably the primary obstacle to achieve an order of magnitude better accuracy. Potential directions for improvement are (1) using ensemble of image and video/audio models, (2) using attention and/or object level models [10], (3) training on more image/video datasets [11] to improve the coverage and domain adaptation.

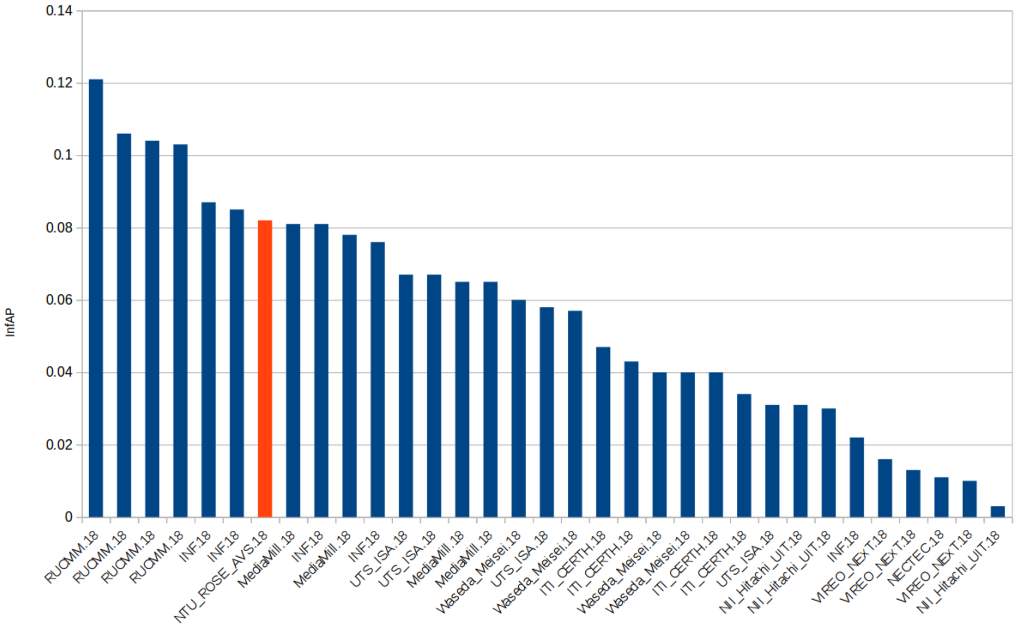

Figure 2: AVS results (inferred average precision, infAP) of fully automatic systems (type F&D) provided by NIST. Mean infAP is 0.056, median infAP is 0.058, maximum infAP is 0.121. Our submission NTU_ROSE_AVS (infAP: 0.082) is shown in orange.

2 Video to Text (VTT)

The video-to-text task consists of 2 sub-tasks which are Description Generation and Matching-and-Ranking [2]. In the Description Generation task, we are given 1921 short videos to generate the corresponding descriptions. In the Matching-and-Ranking task, we are required to rank the given candidate descriptions and return

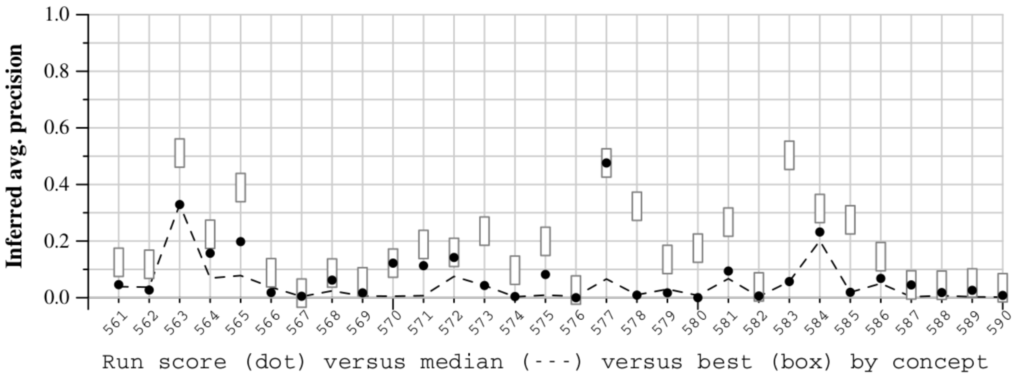

Figure 3: Inferred AP values for each of the 30 test queries for our submission, provided by NIST [2].

a ranked list of the most likely text description corresponding to the video. We trained 2 different models separately for these tasks. For the Description Generation task, we trained a multi-featured video captioning model on a combined video captioning dataset. For the Matching-and-Ranking task, we trained image-based ranking models with 2 different methods on the MS COCO dataset.

2.1 Description Generation

We used the CST-captioning (Consensus-based Sequence Training) [12] as the baseline and adapted it to make it suitable for the task. The framework of the model is shown in Figure 4. The model was trained with the normal cross-entropy loss at the beginning and fine tuned with reinforcement learning.

Figure 5 shows the reinforcement learning processing of Consensus-based Sequence Training (CST). Consensus-based Sequence Training exploits the relationship between the reinforcement loss and cross entropy loss and uses the consensus among training captions as the baseline.

2.1.1 Model Training

We adapted the PyTorch implementation of cst-captioning at `https://github.com/mynlp/cst_captioning`.

Training Dataset. The task did not release the coverage of the generated descriptions. The generated descriptions can be rich and varied. Therefore, we combined MSR-VTT [13] and MSVD [14] datasets to achieve a large domain of description generation.

Video Encoding. For each video, we extracted the image, template, and audio features with ResNet [15], C3D [16], VGGish network [17] separately. We selected 20 frames totally at equal intervals and resized the images to 224 × 224. Then we extracted the image-based features with size is 2048 using FC layer of ResNet101 for the selected frames. The C3D model extracts one feature for every 16 frames in a video. We

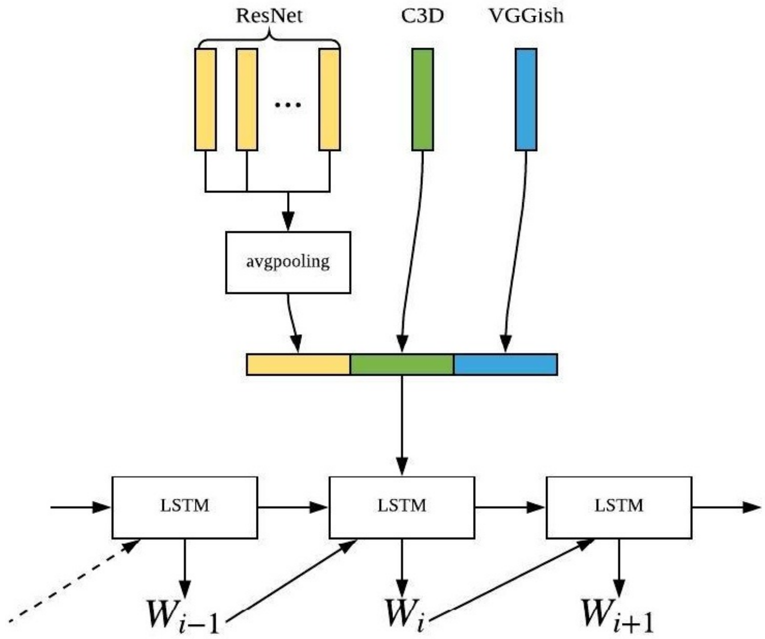

Figure 4: The framework for description generation task.

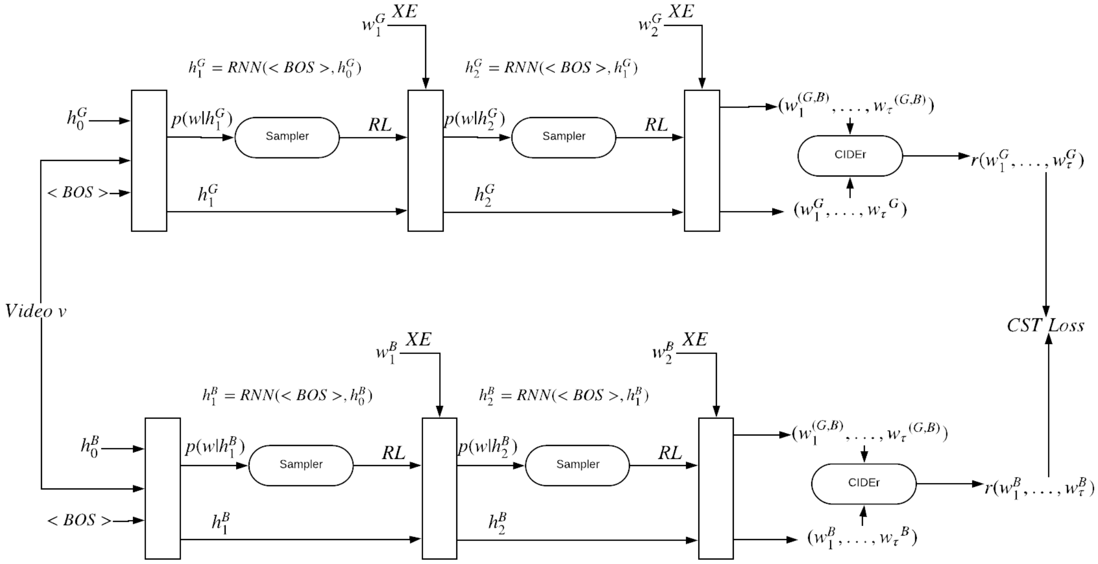
w_1^G XE
w_2^G XE
$h_1^G = RNN(<BOS>, h_0^G)$
$h_2^G = RNN(<BOS>, h_1^G)$
h_0^G
$p(w|h_1^G)$
Sampler
RL
$p(w|h_2^G)$
Sampler
RL
$(w_1^{(G,B)}, \ldots, w_\tau^{(G,B)})$
CIDEr
$r(w_1^G, \ldots, w_\tau^G)$
$<BOS>$
h_1^G
h_2^G
$(w_1^G, \ldots, w_\tau^G)$
$Video\ v$
$CST\ Loss$
w_1^B XE
w_2^B XE
$h_1^B = RNN(<BOS>, h_0^B)$
$h_2^B = RNN(<BOS>, h_1^B)$
h_0^B
$p(w|h_1^B)$
Sampler
RL
$p(w|h_2^B)$
Sampler
RL
$(w_1^{(G,B)}, \ldots, w_\tau^{(G,B)})$
CIDEr
$r(w_1^B, \ldots, w_\tau^B)$
$<BOS>$
h_1^B
h_2^B
$(w_1^B, \ldots, w_\tau^B)$

Figure 5: The reinforcement learning method, Consensus-based Sequence Training (CST), used in this task.

selected continuous 16 frames at intervals of 8 frames. As for the audio-based features, we first separated audio from the video and then extracted the MFCC features. The sample rate we set is 16000 and extracted initial features from 25 ms audio segments with a frame step of 10 ms. Then we further used the initial features as the input of a VGGish network to generate the audio features. Finally, we averaged the image-based and video-based features from different frames into single vectors separately and concatenated all three features together as inputs of the description generator.

Description Generation. We used Adam optimizer with a learning rate of 0.0001 throughout the whole training process. We first trained the model with a mini-batch size of 40 using the cross-entropy loss for 50 epochs. Then we fine-tuned the model with reward-weighted cross entropy loss (WXE) for another 50 epochs. After that, we continued fine-tuning it with CST_MS_SCB method introduced in [12] for 200 epochs with a mini-batch size of 64 to get the final results.

2.1.2 Results and Discussion

We submitted 2 runs using the model and training method introduced above training on different datasets. RUN0 was trained on the dataset consisting of MSVD and MSR-VTT. RUN1 was trained on the MSR-VTT dataset. For each RUN, several metrics are used by NIST [2] to evaluate the generated captions including BLEU, CIDEr, Meteor, and STS. As shown in Figure 6, our result achieved 0.173 in CIDEr ranking 9th among all the submissions and 3rd among best result of each team. STS experimental metric (Semantic Similarity) aims to compute semantic similarity between words/phrases. The metrics consist of a statistical method based on distributional similarity and Latent Semantic Analysis (LSA) and semantic relations extracted from WordNet. In STS metrics, our submissions achieve 0.3650 ranking 4th among all best results of each team, which are shown in Figure 7.

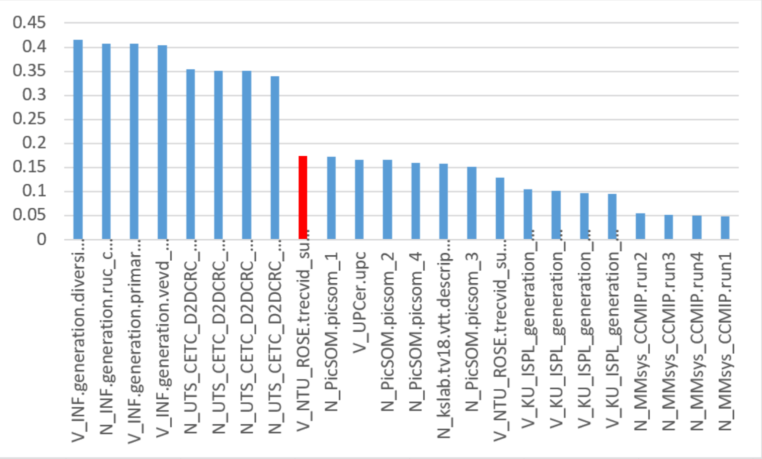

Figure 6: Test results on the CIDEr metrics

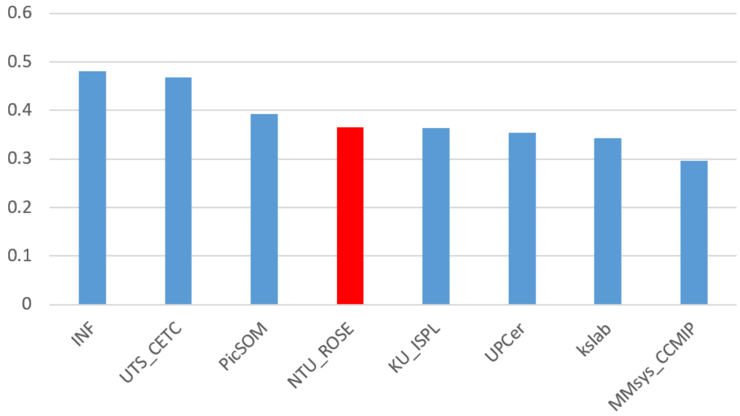

Figure 7: Test results on the STS metrics

Comparing with outstanding results and ground-truth description, out model can generate short descriptions describing the similar topics for the videos. It is mainly due to domain difference in training dataset and test dataset. Observing the target captioning, we can find that the descriptions are variable and some of the words seldom appear in the datasets.

Comparing our submitted RUNs, we find that the model trained on the MSR-VTT performed better than the model trained on the combined dataset. This is probably because MSVD dataset has a larger domain gap. The descriptions of the MSVD dataset is much shorter compared with the test dataset and MSR-VTT dataset. Therefore training on the MSVD dataset will make the model tend to generate shorter sentences instead of detailed ones.

2.2 Matching-and-Ranking

Considering the relatively short video resources and the diverse words in the candidate descriptions, the image-based retrieval method is more suitable for this task. The image-based retrieval method can learn more about the detailed attributes of the entities in the images, which can be used to rank different descriptions. Therefore, we used the VSE++ [4] method for the Matching-and-Ranking subtask.

2.2.1 Algorithm Implementation

For RUN0, we adapted the VSE++ code from the `https://github.com/fartashf/vsepp`. We directly trained this model on the MS COCO captions dataset. During the training, we employed the GloVe [18] embeddings as the pretrained language embedding.

For RUN1, we used a method from our previous work [19]. We apply a cross-modal model to generate the image and video at the same time to improve the retrieval. During the training, the word embedding size is 300 and the size of the joint embedding space is 1024. We used Adam as optimizor in this task. The initial learning rate is 0.0002, and the momentum is 0.9. The mini-batch size is 128. All other parameters are the same as those in the paper. Both models were trained on the MS COCO dataset.

During testing, we considered each frame as an individual image and ranked it with all the candidate descriptions. Then we multiplied all the possibilities of the same video together and sorted from large to small to rank them.

2.2.2 Results and Discussion

Our result is shown in Figure 8, we submitted 2 runs using the model and training method introduced above training on different datasets. RUN1 achieved 0.149 in subset A which ranks 5th among all the teams. For the top 5 ranking results, we can find that some entities and attributes do not match well with the retrieved video. This is mainly because the image-based retrieval focuses on information of the objects and attributes in the image rather than the temporal information such as actions in the video, which is also significant in this task.

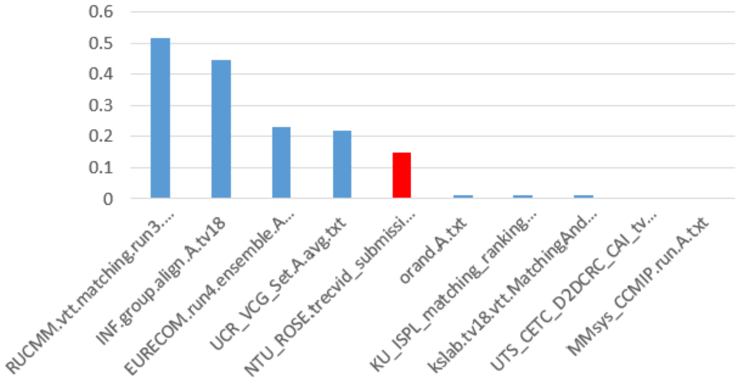

Figure 8: Matching-and-Ranking results on the TRECVID dataset

Acknowledgments

This work was done at the ROSE Lab of Nanyang Technological University, Singapore. The ROSE Lab is supported by the Infocomm Media Development Authority, Singapore. We gratefully acknowledge the support of NVIDIA AI Technology Center for their donation of GPUs used for our research.

References

[1] Jakub Lokoc, Werner Bailer, K.S.B.M.G.A.: On Influential Trends in Interactive Video Retrieval: Video Browser Showdown 2015-2017. IEEE Transactions on Multimedia (2018) 16 pages

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

[3] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: European conference on computer vision, Springer (2014) 740–755

[4] Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2** (2014) 67–78

[5] Vendrov, I., Kiros, R., Fidler, S., Urtasun, R.: Order-embeddings of images and language. arXiv preprint arXiv:1511.06361 (2015)

[6] Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: Improving Visual-Semantic Embeddings with Hard Negatives. In: BMVC. (2018)

[7] OpenCV: OpenCV. `https://opencv.org` (2018)

[8] Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. Volume 96. (1996) 226–231

[9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**(Oct) (2011) 2825–2830

[10] Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked Cross Attention for Image-Text Matching. arXiv preprint arXiv:1803.08024 (2018)

[11] Google: Conceptual Captions Dataset. `https://github.com/google-research-datasets/conceptual-captions` (2018)

[12] Phan, S., Henter, G.E., Miyao, Y., Satoh, S.: Consensus-based sequence training for video captioning. ArXiv e-prints (2017)

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

[14] Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics (2011) 190–200

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778

[16] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision. (2015) 4489–4497

[17] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al.: Cnn architectures for large-scale audio classification. In: Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE (2017) 131–135

[18] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). (2014) 1532–1543

[19] Gu, J., Cai, J., Joty, S., Niu, L., Wang, G.: Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7181–7189

Appendix A Ad-hoc Video Search Queries

The following is the list of 30 queries in TRECVID 2018. The numbers 561 to 590 are the query IDs.

1. 561 Find shots of exactly two men at a conference or meeting table talking in a room
2. 562 Find shots of a person playing keyboard and singing indoors
3. 563 Find shots of one or more people on a moving boat in the water
4. 564 Find shots of a person in front of a blackboard talking or writing in a classroom
5. 565 Find shots of people waving flags outdoors

6. 566 Find shots of a dog playing outdoors
7. 567 Find shots of people performing or dancing outdoors at nighttime
8. 568 Find shots of one or more people hiking
9. 569 Find shots of people standing in line outdoors
10. 570 Find shots of a projection screen
11. 571 Find shots of any type of Christmas decorations
12. 572 Find shots of two or more cats both visible simultaneously
13. 573 Find shots of medical personnel performing medical tasks
14. 574 Find shots of two people fighting
15. 575 Find shots of a person pouring liquid from one container to another
16. 576 Find shots of a person holding his hand to his face
17. 577 Find shots of two or more people wearing coats
18. 578 Find shots of a person in front of or inside a garage
19. 579 Find shots of one or more people in a balcony
20. 580 Find shots of an elevator from the outside or inside view
21. 581 Find shots of a person sitting on a wheelchair
22. 582 Find shots of a person climbing an object (such as tree, stairs, barrier)
23. 583 Find shots of a person holding, talking or blowing into a horn
24. 584 Find shots of a person lying on a bed
25. 585 Find shots of a person with a cigarette
26. 586 Find shots of a truck standing still while a person is walking beside or in front of it
27. 587 Find shots of a person looking out or through a window
28. 588 Find shots of a person holding or attached to a rope
29. 589 Find shots of car driving scenes in a rainy day
30. 590 Find shots of a person where a gate is visible in the background