Supplementary Material of Video Title Generation

Kuo-Hao Zeng[1], Tseng-Hung Chen[1], Juan Carlos Niebles[2], Min Sun[1]

[1]Departmant of Electrical Engineering, National Tsing Hua University
[2]Department of Computer Science, Stanford University
[1]{s103061614@m103,s104061544@m104,sunmin@ee}.nthu.edu.tw
[2]jniebles@cs.stanford.edu

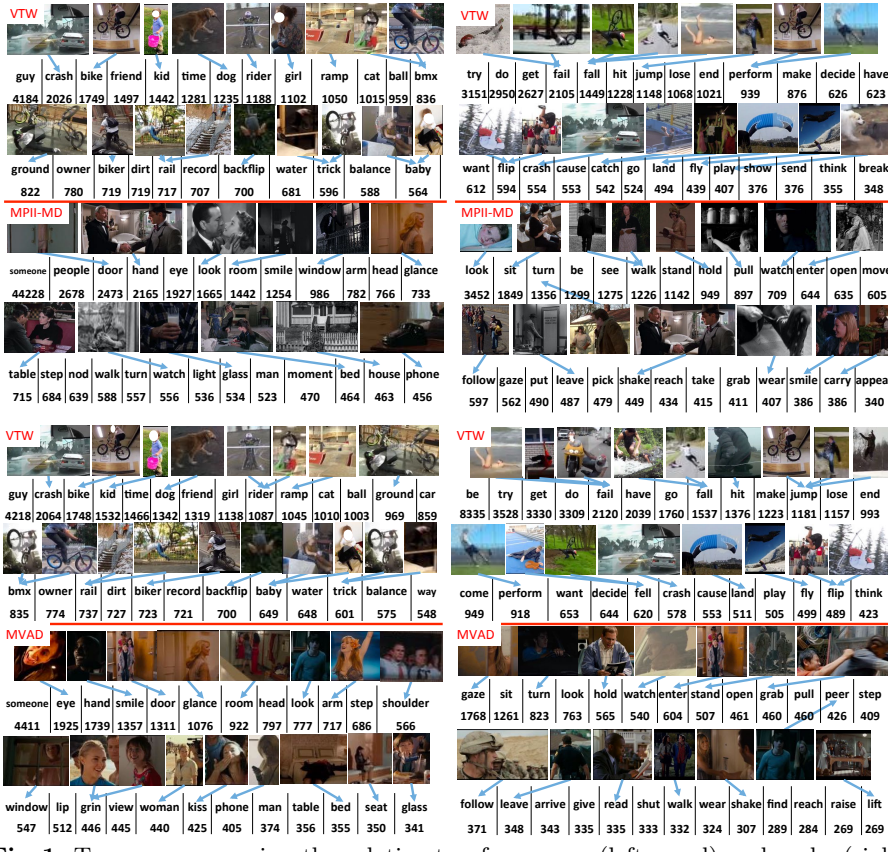

Fig. 1. Top row: comparing the relative top few nouns (left-panel) and verbs (right-panel) between VTW and MPII-MD. dataset [1]. Bottom row: comparing the relative top few nouns (left-panel) and verbs (right-panel) between VTW and M-VAD dataset [2]. The number 4184 for guy (see top-left corner) means that the word "guy" appears 4184 more times in VTW than in MPII-MD.

1 Complementary Vocabulary

The common words are different in these datasets, since VTW consists of user-generated-videos and [1,2] consists of movie clips. We visualize the relative top few nouns and verbs in VTW v.s. MPII-MD. [1] (Fig. 1-Top) and VTW v.s. M-VAD [2] (Fig. 1-Bottom).

2 Dummy Video Observation

For S2VT, we have tried two different types of dummy video observation: (1) single-one, (2) all-zeros. On the whole training and testing dataset, the accuracy of these two types are reported in Table. 1. Since all-zeros outperforms single-one, we select all-zeros as our dummy video observation for S2VT.

VTW	S2VT [3] (%)					
Variant	B@1	B@2	B@3	B@4	MET.	CIDEr
single-one	11.7	4.9	2.0	0.9	5.6	21.7
all-zeros	11.0	4.7	2.3	1.3	6.0	22.8

Table 1. Video captioning performance of different types of dummy video observation on a whole training and testing dataset.

3 Testing-Word-Count-in-Training

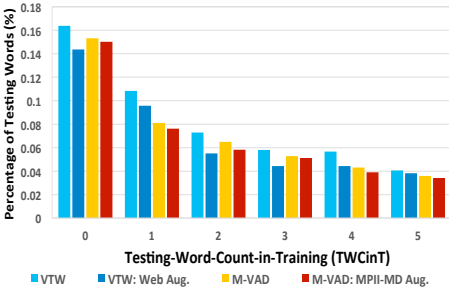

Fig. 2. We show the percentage (y-axis) of unique words in the test set that appears zero to five times (x-axis) in training from left to right. We compare these statistics on our VTW dataset before (VTW-title) and after (VTW-Aug.) sentence augmentation. The same comparison is done for M-VAD [2] dataset before (M-VAD) and after (M-VAD-Aug.) sentence augmentation.

We report the Testing-Word-Count-in-Training (TWCinT) statistics on VTW and M-VAD dataset before and after sentence augmentation in Table. 2.

4 Our TensorFlow Implementation

We reimplement S2VT [3] and SA [4] using TensorFlow [5]. We follow [3] to implement sequence to sequence translation between video observations and corresponding video title as Fig. 3-Top. For simplification, each LSTM block in the

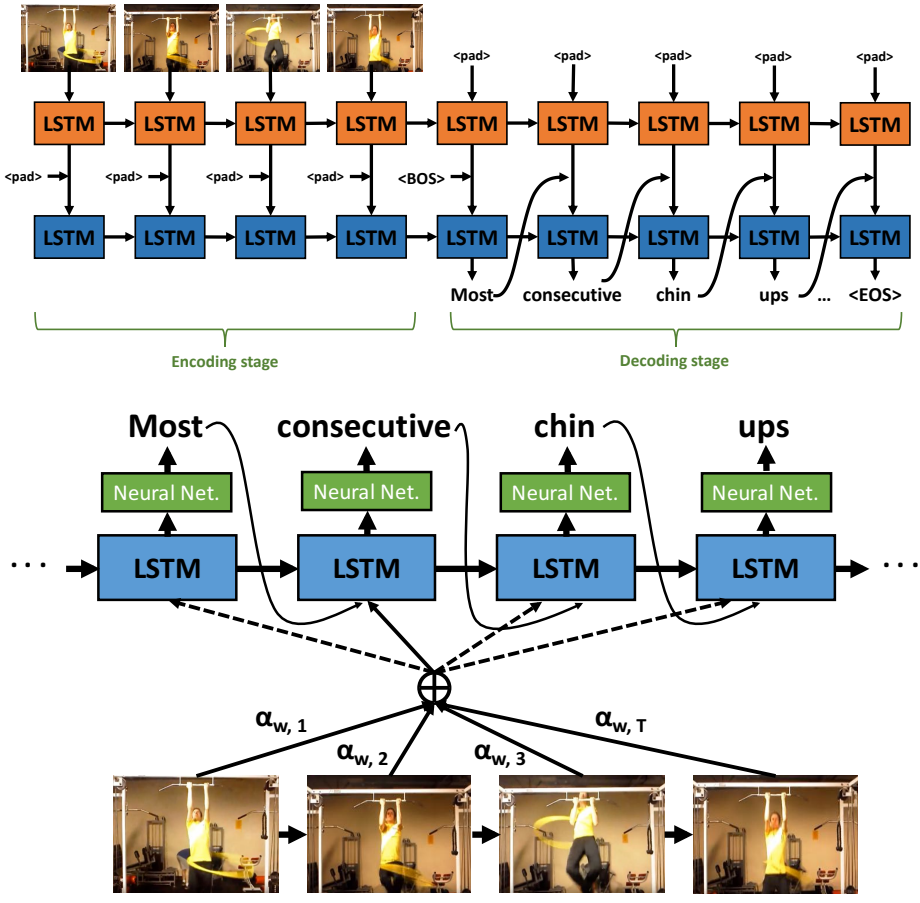

Fig. 3. Top figure: it is our implementation of [3]. Note that $< pad >$ means zero-padding, $< BOS >$ means the start of the sentence, and $< EOS >$ means the end of the sentence. Bottom figure: it is our implementation of [4]. Note that for each word w, we recompute soft-attention scores $\{\alpha_{w,t}\}_t$ and utilize those scores $\{\alpha_{w,t}\}_t$ to weighted sum over all clips (indexed by t).

figure is implemented by the standard module in TensorFlow [5]. In order to separate visual embedding feature and language embedding feature, we follow [3] to use the all zero vector to implicitly show the absence of each feature. $< BOS >$ and $< EOS >$ are special one-hot vectors indicating the start of the sentence and the end of the sentence, respectively. Note that while only a visual feature channel is shown, we follow [3] to replicate the same architecture with different learnable weights and bias for VGG [6] and C3D [7]. Then, a late-fusion layer is used to combine the results from different feature sources. On the other hand, we follow [4] to implement soft-attention model as Fig. 3-Bottom. A single layer LSTM network is used to generate predicted word at each time step. Note

that different from the S2VT model, we follow [4] to early fuse appearance and motion features by concatenating them.

5 Highlight Detector

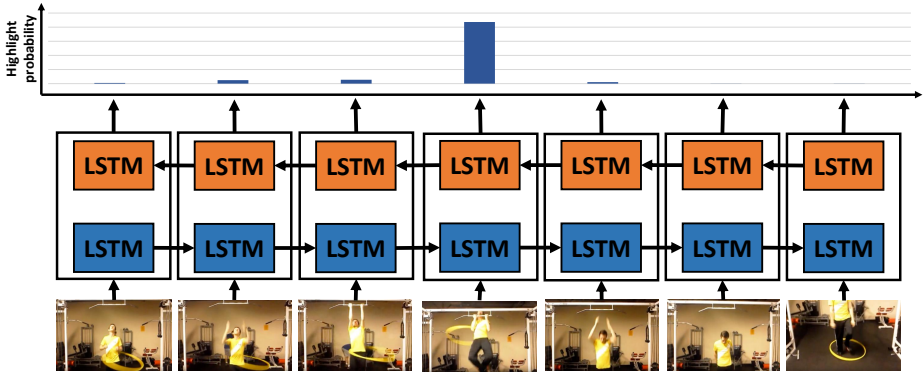

Fig. 4. It is the implementation of our highlight detector. We use a bidirectional single-layer LSTM model with binary classification strategy to discriminate highlight or non-highlight for each video. Higher highlight probability means higher chance to be detected as highlight clip.

For highlight detector, we implement a bidirectional single-layer LSTM model with binary classification strategy as Fig. 4. Each clip can be discriminated as highlight or non-highlight by a softmax function. In the testing phase, our highlight detector could predict highlight probability for each testing clip[1]. In order to verify our setting, we train on 2000 training videos (14.2% of full training set), select best model on 300 validation videos (15% of full validation set), and test on whole testing videos. Initially, our highlight detector achieves 54.2% mean average precision (mAP). When the highlight sensitive video captioner is trained, we can obtain clip-wise captioning loss by directly feeding each clip into the captioner. Next, we take the highlight score as the negative of the captioning loss in the unlabeled training set (12100 videos). We propose to use this clip-wise highlight score to select additional highlight and non-highlight clips to retrain our highlight detector. However, since our high capacity bidirectional RNN model can easily fit to incorrectly selected highlight and non-highlight clips, we only select a subset of highly confident videos for retraining. We propose to select videos with a clear single highlight clip for retraining. In detail, we first subtract the score by the minimum score in each video. Then, we ignore videos having any local maximum score higher than 10% of global maximum score. All remained videos are selected into the training set. In each newly selected training video, we

[1] Higher highlight probability means higher chance to be detected as highlight.

treat the clip having the global maximum score be the highlight, ignore the two clips beside the highlight, and all remaining clips as non-highlights. Finally, we gain 2731 more training videos and the number of total labeled training videos increase to 4731. After retraining highlight detector using new training videos, the final highlight detector has 58.3% mAP on the highlight detection task.

6 Human Evaluation

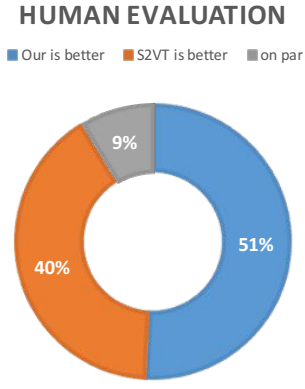

Fig. 5. The distribution of human evaluation. The blue segment represents that our method (HL+Web Aug.) is better, the orange segment represents that S2VT baseline method is better, and the grey segment represents on par. Our titles outperform S2VT titles by 11% more (51% − 40%) and human judges decide that 59.5% of our sentences are on par or better than the S2VT sentences.

We conduct a human evaluation comparing S2VT baseline method and our method (HL+Web Aug.). Before conducting large-scale human evaluation, we first randomly sample a few sample titles pairs (S2VT v.s. ours) and ask human judges to make hard decision from the following three options: "S2VT is better", "ours is better", and "on par". We found that this is a difficult hard decision, when both titles are partially relevant to the video content but in different ways. To avoid introducing potential random hard decision by the human judges, we sort the title pairs according to the smaller METEOR between S2VT title and our title as follows,

$$minMETEOR = min(METEOR._{S2VT}, METEOR._{HL+WebAug.}). \quad (1)$$

Then, we choose the top 1000 videos with higher minMETEOR scores for human evaluation. We ask seven subjects to conduct blind test which means that all subjects do not know which sentence is predicted by S2VT or our method. In the end, among 1000 videos, 508 (51%) videos are selected as "ours is better", 405 (40%) videos are selected as "S2VT is better", and the remained 87 (9%) videos are selected as "on par" (Fig. 5). As a result, our titles outperform S2VT

titles by 11% more (51% − 40%), and human judges decide that 59.5% of our sentences are on par or better than the S2VT sentences.

References

1. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR. (2015)
2. Torabi, A., Pal, C.J., Larochelle, H., Courville, A.C.: Using descriptive video services to create a large data source for video annotation research. arXiv:1503.01070 (2015)
3. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence - video to text. In: ICCV. (2015)
4. Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville., A.: Describing videos by exploiting temporal structure. In: ICCV. (2015)
5. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
7. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. (2015)