Supplementary Material: Multi-Task Video Captioning with Video and Entailment Generation

Ramakanth Pasunuru and **Mohit Bansal**
UNC Chapel Hill
{ram, mbansal}@cs.unc.edu

1 Experimental Setup

1.1 Datasets

1.1.1 Video Captioning Datasets

YouTube2Text or MSVD The Microsoft Research Video Description Corpus (MSVD) or YouTube2Text (Chen and Dolan, 2011) is used for our primary video captioning experiments. It has 1970 YouTube videos in the wild with many diverse captions in multiple languages for each video. Caption annotations to these videos are collected using Amazon Mechanical Turk (AMT). All our experiments use only English captions. On average, each video has 40 captions, and the overall dataset has about $80,000$ unique video-caption pairs. The average clip duration is roughly 10 seconds. We used the standard split as stated in Venugopalan et al. (2015), i.e., 1200 videos for training, 100 videos for validation, and 670 for testing.

MSR-VTT MSR-VTT is a recent collection of $10,000$ video clips of 41.2 hours duration (i.e., average duration of 15 seconds), which are annotated by AMT workers. It has $200,000$ video clip-sentence pairs covering diverse content from a commercial video search engine. On average, each clip is annotated with 20 natural language captions. We used the standard split as provided in (Xu et al., 2016), i.e., $6,513$ video clips for training, 497 for validation, and $2,990$ for testing.

M-VAD M-VAD is a movie description dataset with $49,000$ video clips collected from 92 movies, with the average clip duration being 6 seconds. Alignment of descriptions to video clips is done through an automatic procedure using Descriptive Video Service (DVS) provided for the movies. Each video clip description has only 1 or 2 sentences, making most evaluation metrics (except paraphrase-based METEOR) infeasible. Again, we used the standard train/val/test split as provided in Torabi et al. (2015).

1.1.2 Video Prediction Dataset

For our unsupervised video representation learning task, we use the UCF-101 action videos dataset (Soomro et al., 2012), which contains $13,320$ video clips of 101 action categories and with an average clip length of 7.21 seconds each. This dataset suits our video captioning task well because both contain short video clips of a single action or few actions, and hence using future frame prediction on UCF-101 helps learn more robust and context-aware video representations for our short clip video captioning task. We use the standard split of $9,500$ videos for training (we don't need any validation set in our setup because we directly tune on the validation set of the video captioning task).

1.2 Pre-trained Visual Frame Features

For the three video captioning datasets (Youtube2Text, MSR-VTT, M-VAD) and the unsupervised video prediction dataset (UCF-101), we fix our sampling rate to $3fps$ to bring uniformity in the temporal representation of actions across all videos. These sampled frames are then converted into features using several state-of-the-art pre-trained models on ImageNet (Deng et al., 2009) – VGGNet (Simonyan and Zisserman, 2015), GoogLeNet (Szegedy et al., 2015; Ioffe and Szegedy, 2015), and Inception-v4 (Szegedy et al., 2016). For VGGNet, we use its $fc7$ layer features with dimension 4096. For GoogLeNet and Inception-v4, we use the layer before the fully connected layer with dimensions 1024 and 1536, respectively. We follow standard preprocessing and convert all the natural language descriptions to lower case and tokenize the sentences and remove punctuations.

2 Training Details

In all of our experiments, we tune all the model hyperparameters on validation (development) set of the corresponding dataset. We consider the following short hyperparameters ranges and tune lightly on: LSTM-RNN hidden state size - $\{256, 512, 1024\}$; learning rate in the range $[10^{-5}, 10^{-2}]$ with uniform intervals on a log-scale; weight initializations in the range $[-0.1, 0.1]$ and mixing ratios in the range 1:$[0.01, 3]$ with uniform intervals on a log-scale. We use the following settings in all of our models (unless otherwise specified in a subsection below): we unroll video encoder/decoder LSTM-RNNs to 50 time steps and language encoder/decoder LSTM-RNNs to 30 time steps. We use a 1024-dimension LSTM-RNN hidden state size. We use 512-dimension vectors to embed frame level visual features and word vectors. These embedding weights are learned during the training. We use the Adam optimizer (Kingma and Ba, 2015) with default coefficients and a batch size of 32. We apply a dropout with probability 0.5 to the vertical connections of LSTM (Zaremba et al., 2014) to reduce overfitting.

2.1 Video Captioning on YouTube2Text

2.1.1 Baseline and Attention Models

Our primary baseline model (Inception-v4, attention, ensemble) uses a learning rate of 0.0001 and initializes all its weights with a uniform distribution in the range $[-0.05, 0.05]$.

2.1.2 Multi-Task with Video Prediction (1-to-M)

In this model, the video captioning and unsupervised video prediction tasks share their encoder LSTM-RNN weights and image embeddings in a one-to-many multi-task setting. We again use a learning rate of 0.0001 and initialize all the learnable weights with a uniform distribution in the range $[-0.05, 0.05]$. Two important hyperparameters tuned (on the validation set of captioning datasets) are the ratio of encoder vs decoder frames for video prediction on UCF-101 (where we found that 80% of frames as input and 20% for prediction performs best); and the mini-batch mixing ratio between the captioning and video prediction tasks (where we found $100 : 200$ works well).

2.1.3 Multi-Task with Entailment Generation (M-to-1)

In this model, the video captioning and entailment generation tasks share their language decoder LSTM-RNN weights and word embeddings in a many-to-one multi-task setting. We again use a learning rate of 0.0001. All the trainable weights are initialized with a uniform distribution in the range $[-0.08, 0.08]$. We observe that a mixing ratio of $100 : 50$ (between the captioning and entailment generation tasks) alternating mini-batches works well here.

2.1.4 Multi-Task with Video and Entailment Generation (M-to-M)

In this many-to-many, three-task model, the video encoder is shared between the video captioning and unsupervised video prediction tasks, and the language decoder is shared between the video captioning and entailment generation tasks. We again use a learning rate of 0.0001. All the trainable weights are initialized with a uniform distribution in the range $[-0.08, 0.08]$. We found that a mixing ratio of $100 : 100 : 50$ alternative mini-batches of video captioning, unsupervised video prediction, and entailment prediction works best.

2.2 Video Captioning on MSR-VTT

We also evaluate our many-to-many multi-task model on other video captioning datasets. For MSR-VTT, we train the model again using a learning rate of 0.0001. All the trainable weights are initialized with a uniform distribution in the range $[-0.05, 0.05]$. We found that a mixing ratio of $100 : 20 : 20$ alternative mini-batches of video captioning, unsupervised video prediction, and entailment prediction works best.

2.3 Video Captioning on M-VAD

For the M-VAD dataset, we use 512 dimension hidden vectors for the LSTMs to reduce overfitting. We initialize the LSTM weights with a uniform distribution in the range $[-0.1, 0.1]$ and all other weights with a uniform distribution in the range $[-0.05, 0.05]$. We use a learning rate of 0.001. We found a mixing ratio of $100 : 5 : 5$ alternative mini-batches of video captioning, unsupervised video prediction, and entailment prediction works best.

MULTI-TASK WITH VIDEO PREDICTION (1-TO-M)

Ground-truth Video Captions	Baseline	Multi-Task (1-to-M)
a man drinks a glass of water a man drinks something	a man is eating something	a man is drinking something
a man scores when playing basketball young man dribbles and throws basketball	a man is playing with a ball	a man is playing a basketball
a person cuts a piece of blue paper a woman is cutting a paper in square by a scissor	a man is playing with a board	a man is cutting a paper
a man is cutting meat with axe a man is chopping a chicken	a man is cooking	a man is cutting a piece of meat
a woman is slicing onions a woman is chopping an onion	a woman is slicing a vegetable	a woman is slicing an onion
a train is going down the track near a shore a high speed train is running down the track	a train is playing	a train is going on a track

MULTI-TASK WITH ENTAILMENT GENERATION (M-TO-1)

Ground-truth Video Captions	Baseline	Multi-Task (M-to-1)
a cat is walking on the ground a cat is sneaking through some grass	a cat is playing with a cat	a cat is playing
a woman is applying eye liner a woman applies makeup to her eye brows	a woman is talking	a woman is doing makeup
a baby tiger is playing the tiger is playing	a tiger is playing with a tiger	a tiger is playing
a man and woman are driving on a motorcycle the man gave the woman a ride on the motorcycle	a man is riding a song	a man is riding a motorcycle
a boy is walking on a treadmill a man exercising with a baby	a man is cleaning the floor	a man is exercising
a puppy is playing on a sofa a puppy is running around on a sofa	a dog is playing with a dog	a puppy is playing

Table 1: Examples showing cases where our one-to-many and many-to-one multi-task video-captioning models are better than the baseline.

2.4 Entailment Generation

Here, we use video captioning to in turn help improve entailment generation results. We use the same hyperparameters for both the baseline and the multi-task model (Sec. 5.3 and Table 4). We use a learning rate of 0.001. All the trainable weights are initialized with a uniform distribution in the range $[-0.08, 0.08]$. We found a mixing ratio of $100 : 20$ alternate mini-batches training of entailment generation and video captioning to perform best.

3 Analysis

In Sec. 5.5 of the main paper, we discussed examples comparing the generated captions of the final many-to-many multi-task model with those of the baseline. Here, we also separately compare our one-to-many (video prediction based) and many-to-one (entailment generation based) multi-task models with the baseline. As shown in Table 1, our one-to-many multi-task model better identifies the actions and objects in comparison to the baseline, because the video prediction task helps it learn better context-aware visual representations, e.g., “a man is eating something” vs. “a man is drinking something” and “a woman is slicing a vegetable” vs. “a woman is slicing an onion”.

On the other hand, the many-to-one multi-task (with entailment generation) seems to be stronger at generating a caption which is a logically-implied entailment of a ground-truth caption, e.g., “a cat is playing with a cat” vs. “a cat is playing” and “a woman is talking” vs “a woman is doing makeup” (see Table 1).

References

David L Chen and William B Dolan. 2011. 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, pages 190–200.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. IEEE, pages 248–255.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* .

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. In *CoRR*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*. pages 1–9.

Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070* .

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *CVPR*. pages 4534–4542.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*. pages 5288–5296.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .