Supplementary Material: Reinforced Video Captioning with Entailment Rewards

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

1 Attention-based Baseline Model (Cross-Entropy)

Our attention baseline model is similar to the Bahdanau et al. (2015) architecture, where we encode input frame level video features to a bi-directional LSTM-RNN and then generate the caption using a single layer LSTM-RNN, with an attention mechanism. Let $\{f_1, f_2, ..., f_n\}$ be the frame-level features of a video clip and $\{w_1, w_2, ..., w_m\}$ be the sequence of words forming a caption. The distribution of words at time step t given the previously generated words and input video frame-level features is given as follows:

$$p(w_t|w_{1:t-1}, f_{1:n}) = softmax(W^T h_t^d) \quad (1)$$

where w_t and h_t^d are the generated word and hidden state of the LSTM decoder at time step t. W^T is the projection matrix. h_t^d is given as follows:

$$h_t^d = S(h_{t-1}^d, w_{t-1}, c_t) \quad (2)$$

where S is a non-linear function. h_{t-1}^d and w_{t-1} are previous time step's hidden state and generated word. c_t is a context vector which is a linear weighted combination of the encoder hidden states h_i^e, given by $c_t = \sum \alpha_{t,i} h_i^e$. These weights $\alpha_{t,i}$ act as an attention mechanism, and are defined as follows:

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{k=1}^{n} exp(e_{t,k})} \quad (3)$$

where the attention function $e_{t,i}$ is defined as:

$$e_{t,i} = w^T tanh(W_a h_i^e + U_a h_{t-1}^d + b_a) \quad (4)$$

where w, W_a, Ua, and b_a are trained attention parameters. Let θ be the model parameters and $\{w_1^*, w_2^*, ..., w_m^*\}$ be the ground-truth word sequence, then the cross entropy loss optimization function is defined as follows:

$$L(\theta) = -\sum_{i=1}^{m} \log(p(w_t^*|w_{1:t-1}^*, f_{1:n})) \quad (5)$$

2 Reinforcement Learning (Policy Gradient)

Traditional video captioning systems minimize the cross entropy loss during training, but typically evaluated using phrase-matching metrics: BLEU, METEOR, CIDEr, and ROUGE-L. This discrepancy can be addressed by directly optimizing the non-differentiable metric scores using policy gradients p_θ, where θ represents the model parameters. In our captioning system, our baseline attention model acts as an agent and interacts with its environment (video and caption). At each time step, the agent generates a word (action), and the generation of the end-of-sequence token results in a reward r to the agent. Our training objective is to minimize the negative expected reward function given by:

$$L(\theta) = -\mathbb{E}_{w^s \sim p_\theta}[r(w^s)] \quad (6)$$

where $w^s = \{w_1^s, w_2^s, ..., w_m^s\}$, and w_t^s is the word sampled from the model at time step t. Based on the REINFORCE algorithm (Williams, 1992), the gradients of the non-differentiable, reward-based loss function can be computed as follows:

$$\nabla_\theta L(\theta) = -\mathbb{E}_{w^s \sim p_\theta}[r(w^s) \nabla_\theta \log p_\theta(w^s)] \quad (7)$$

The above gradients can be approximated from a single sampled word sequence w^s from p_θ as follows:

$$\nabla_\theta L(\theta) \approx -r(w^s) \nabla_\theta \log p_\theta(w^s) \quad (8)$$

However, the above approximation has high variance because of estimating the gradient with a single sample. Adding a baseline estimator reduces this variance (Williams, 1992) without changing the expected gradient. Hence, Eqn: 8 can be rewritten as follows:

$$\nabla_\theta L(\theta) \approx -(r(w^s) - b_t) \nabla_\theta \log p_\theta(w^s) \quad (9)$$

where b_t is the baseline estimator, where b_t can be a function of θ or time step t, but not a function of w^s. In our model, baseline estimator is a simple linear regressor with hidden state of the decoder h_t^d at time step t as the input. We stop the back propagation of gradients before the hidden states for the baseline bias estimator. Using the chain rule, loss function can be written as:

$$\nabla_\theta L(\theta) = \sum_{t=1}^{m} \frac{\partial L}{\partial s_t} \frac{\partial s_t}{\partial \theta} \tag{10}$$

where s_t is the input to the *softmax* layer, where $s_t = W^T h_t^d$. $\frac{\partial L}{\partial s_t}$ is given by (Zaremba and Sutskever, 2015) as follows:

$$\frac{\partial L}{\partial s_t} \approx (r(w^s) - b_t)(p_\theta(w_t|h_t^d) - 1_{w_t^s}) \tag{11}$$

The overall intuition behind this gradient formulation is: if the reward $r(w^s)$ for the sampled word sequence w^s is greater than the baseline estimator b_t, the gradient of the loss function becomes negative, then model encourages the sampled distribution by increasing their word probabilities, otherwise the model discourages the sampled distribution by decreasing their word probabilities.

3 Experimental Setup

3.1 MSR-VTT Dataset

MSR-VTT is a diverse collection of $10,000$ video clips (41.2 hours of duration) from a commercial video search engine. Each video has 20 human annotated reference captions collected through Amazon Mechanical Turk (AMT). We use the standard split as provided in (Xu et al., 2016), i.e., 6513 for training, 497 for testing , and remaining for testing. For each video, we sample at $3fps$ and we extract Inception-v4 (Szegedy et al., 2016) features from these sampled frames and we also remove all the punctuations from the text data.

3.2 YouTube2Text Dataset

We also evaluate our models on YouTube2Text dataset (Chen and Dolan, 2011). This dataset has 1970 video clips and each clip is annotated with an average of 40 captions by humans. We use the standard split as given in (Venugopalan et al., 2015), i.e., 1200 clips for training, 100 for validation and 670 for testing. We do similar preprocessing as the MSR-VTT dataset.

3.3 Automatic Evaluation Metrics

We use several standard automated evaluation metrics: METEOR (Denkowski and Lavie, 2014), BLEU-4 (Papineni et al., 2002), CIDEr-D (Vedantam et al., 2015), and ROUGE-L (Lin, 2004). We use the standard Microsoft-COCO evaluation server (Chen et al., 2015).

3.4 Human Evaluation

Apart from the automatic metrics, we also present human evaluation comparing the CIDEnt-reward model with the CIDEr-reward model, esp. because the automatic metrics cannot be trusted solely. Human evaluation uses *Relevance* and *Coherence* as the comparison metrics. Relevance is about how related is the generated caption w.r.t. the content of the video, whereas coherence is about the logic, fluency, and readability of the generated caption.

4 Training Details

All the hyperparameters are tuned on the validation set. For each of our main models (baseline, CIDEr and CIDEnt), we report the results on a 5-avg-ensemble, where we run the model 5 times with different initialization random seeds and take the average probabilities at each time step of the decoder during inference time. We use a fixed size step LSTM-RNN encoder-decoder, with encoder step size of 50 and decoder step size of 16. Each LSTM has a hidden size of 1024. We use Inception-v4 features as video frame-level features. We use word embedding size of 512. Also, we project down the 1536-dim image features (Inception-v4) to 512-dim.

We apply dropout to vertical connections as proposed in Zaremba et al. (2014), with a value 0.5 and a gradient clip size of 10. We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001 for baseline cross-entropy loss. All the trainable weights are initialized with a uniform distribution in the range $[-0.08, 0.08]$. During the test time inference, we use beam search of size 5. All our reward-based models use mixed loss optimization (Paulus et al., 2017; Wu et al., 2016), where we train the model based on weighted (γ) combination of cross-entropy loss and reinforcement loss. For MSR-VTT dataset, we use $\gamma = 0.9995$ for our CIDEr-RL model and $\gamma = 0.9990$ for our CIDEnt-RL model. For YouTube2Text/MSVD dataset, we use $\gamma = 0.9985$ for our CIDEr-RL model

Ground truth: A man is playing a violin.
A man is playing the violin on stage.
Baseline-XE: A man is playing the drums.
CIDEr-RL: A man is playing a guitar.
CIDEnt-RL: A man is playing a violin.

Ground truth: Two men are wrestling.
Two guys are wrestling in a competition.
Baseline-XE: A group of people are playing a game.
CIDEr-RL: A man is playing a wrestling.
CIDEnt-RL: Two men are wrestling.

Ground truth: A person is playing a video game.
Someone is playing video game.
Baseline-XE: A man is riding a motorcycle.
CIDEr-RL: A man is talking about a plane.
CIDEnt-RL: A person is playing a video game.

Figure 1: Output examples where our CIDEnt-RL model produces better entailed captions than the phrase-matching CIDEr-RL model, which in turn is better than the baseline cross-entropy model.

and $\gamma = 0.9990$ and for our CIDEnt-RL model. The learning rate for the mixed-loss optimization is 1×10^{-5} for MSR-VTT, and 1×10^{-6} for YouTube2Text/MSVD. The λ hyperparameter in our CIDEnt reward formulation (see Sec. 4 in main paper) is roughly equal to the baseline cross-entropy model's score on that metric, i.e., $\lambda = 0.45$ for MSR-VTT CIDEnt-RL model and $\lambda = 0.75$ for YouTube2Text/MSVD CIDEnt-RL model.

5 Analysis

Figure 1 shows several examples where our CIDEnt-reward model produces better entailed captions than the ones generated by the CIDEr-reward model. This is because the CIDEr-style captioning metrics achieve a high score even when the generation does not exactly entail the ground truth but is just a high phrase overlap. This can obviously cause issues by inserting a single wrong word such as a negation, contradiction, or wrong action/object. On the other hand, our entailment-enhanced CIDEnt score is only high when both CIDEr and the entailment classifier achieve high scores. The CIDEr-RL model, in turn, produces better captions than the baseline cross-entropy model, which is not aware of sentence-level matching at all.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

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.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *EACL*.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 workshop*. volume 8.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*. pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* .

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

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*. pages 4566–4575.

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.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

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 and Ilya Sutskever. 2015. Reinforcement learning neural turing machines. *arXiv preprint arXiv:1505.00521* 362.

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