UCI University of California, Irvine

Deep Learning for Video Captioning: A Review

Shaoxiang Chen, Ting Yao, and Yu-Gang Jiang
(IJCAI-19)

Presented by: Saman Porhemmat

Fall 2019

Roadmap

- Definition of Video Captioning
- Problem Formulation
- Video Representation
- Caption Generation
- Datasets & Evaluation
- Conclusion & Future Work

Definition of Video Captioning

- Describe the visual content of a video using a natural-language utterance.
- Two main components
  - Visual representation
  - Caption generation
- Practical Applications:
  - Video indexing or video retrieval
  - Helping people with visual impairments (visual signal ⟶ information)
- Background:
  - Started approximately around 2002
    - Hand-crafted features for detecting visual concepts.
    - Pre-defined templates for generating sentences.
  - Current approaches utilize deep learning (encoder-decoder paradigm)
    - Encoder is used for learning video representation (from multimodal features).
    - Decoder is used for generating sentences based on the learned representation.

Problem Formulation

- We use the following representation for an input video: $V = \{f_1, f_2, \ldots, f_N\}$

  f is a frame, and *N* is the length of frame sequence.

- The goal of video captioning is to generate a sentence (sequence of words). This can be represented as:

  Where *y* is a word and *T* is the length of the sequence. $Y = \{y_1, y_2, \ldots, y_T\}$

- In order to model the video, we need to extract features: $\mathcal{F} = \{F_V, F_M, F_A, F_S\}$

  We have different types of features: semantic, audio, motion, and visual.

- We can use a condensed representation: $\mathcal{F} = f_{feat}(V)$

- Or we can condense it even more: $F_t = f_{aggr}(\mathcal{F}, s_t)$
  - *st* is a state vector (model's state when generating the *t*-th word)
  - Ft is the aggregated feature

Problem Formulation

f_{feat} and f_{aggr} create the encoder for us.

Now, the decoder or our language model gets F_t and s_t and predicts the distribution of the word y_t. So:

$$p_t = f_{lang}(F_T, s_t)$$

f is our language model, a Recurrent Neural Network (RNN). The final prediction of Y can be written as:

$$\{p_1, p_2, \ldots, p_T\}$$

Video Representation

In order to represent the video correctly, we need to perform two steps:

- Multimodal Feature Extraction
  - Visual
  - Motion
  - Audio
  - Semantic

- Feature Aggregation
  - Temporal Attention
  - Spatial Attention
  - Multimodal Feature Fusion

❖ NOTE: This paper focuses on deep learning.

Multimodal Feature Extraction

- Visual
  - The most important feature for understanding the video
  - Convolutional Neural Networks (CNNs) are heavily used.
    - Layers that are deeper capture simple features such as color gradients and edges.
    - Higher layers combine simple features into more complex features.
  - Popular CNNs for video captioning are the VGG and Inception Networks.

- Motion
  - Crucial for capturing the actions and temporal interactions in video.
  - This type of feature complements the static visual appearance.
  - 3D CNNs learn spatiotemporal features via a sequence of frames. Can handle both motion and appearance.

Multimodal Feature Extraction Cont'd

- Audio
  - Helps with distinguishing events. “Person talking to the phone” and “Person listening to music using a phone”.
  - The Mel Frequency Cepstral Coefficients (MFCC) technique is used to get audio features.
    - MFC: used for representing the short-term power spectrum of a sound.
    - This is based on a linear cosine transform of a log power spectrum on a mel scale of frequency.
  - Video captioning studies usually utilize bag-of-audio-words to acquire a fixed-length audio feature
    - Inspired by bag-of-words, audio features are represented in the form of compact ‘audio words’, whereby each word corresponds to a combination of acoustic features.
- Semantic
  - Semantic features capture semantic contents in videos.
  - In recent research, it has been shown that video-level category information can enhance video captioning.

Feature Aggregation

Video features are acquired from several modalities. Hence, they are sequences of variable length. We want to aggregate them into a fixed-length representation.

Simplest way to aggregate a feature sequence is via an LSTM/GRU model. But there are two problems: - vanishing gradient - each feature contributes the same

- Temporal Attention:
  - Learns to dynamically assign weights to each feature in the sequence such that the decoder can pay more attention to relevant features.
  - The decoder and each feature is directly connected by a weighted path. Reduces the length of gradient flow.
  - hLSTM (2017) is an improved temporal attention mechanism. Decoder depends less on visual features when generating non-visual words. Instead relies on the state of the language model.

Feature Aggregation Cont'd

- Spatial Attention
  - Idea: Different regions of a video frame contribute differently to the final word prediction.
  - These methods learn the importance of regions
  - Temporal attention can be applied spatially if regions are treated sequentially:
    - MAM-RNN (2017): When computing spatial attention weights for a certain frame, the model also considers the attention weights from previous frame.
  - SAM (2018) is a model that distinguishes foreground from background automatically.

- Multimodal Feature Fusion
  - This is a research direction that is rarely explored.
  - In the MMVD (2016) network, we simply concatenate features from multiple modalities and pass the concatenated features to the input of the decoder.
  - In AttentionFusion (2017) and MA-LSTM (2017), the authors apply temporal attention to different streams (visual, audio, motion) of features. This allows each modality to contribute differently to caption generation.

Caption Generation

Given that we can generate word probabilities at each time step $\{p_1, p_2, \ldots, p_T\}$ and that we have ground truth caption $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T\}$ our objective is to maximize the log-likelihood of all the ground truth words:

$$\max_{\theta} \sum_{t=1}^{T} p_t(\hat{y}_t)$$

θ is all the learnable parameters of the captioning model.

Two major problems with this objective function:

- Objective Mismatch
- Inadequate to train a good language model (captioning datasets have a much smaller corpus compared to NLP datasets.

Auxiliary Semantic Supervision

To improve captioning quality, we can make sentence semantics consistent with visual contents by enforcing such consistency.

Representative methods project the encoded visual feature vector and the averaged sentence embedding vector onto a common space. Then, they add an optimization term to minimize their distance, which is jointly optimized with the captioning objective.

In Semantic Attention (SA) the model's attention cover all the semantic concepts in an image. To do this, SA first detects semantic concepts in image and then applies temporal attention to the concepts at each word generation step.

Auxiliary Semantic Supervision Cont'd

In contrast, M&M TGM [Chen et al., 2017] uses predicted semantic topics to guide the learning of video captioning model. A topic mining module first mines topics from the training descriptions by clustering, then it is used as the teacher to train a topic predictor. For caption generation, the predicted topics are fed to an extended LSTM decoder with a set of topic-dependent weight matrices, which works as an ensemble of several topic-aware decoders

Addressing Objective Mismatch

The cause of the objective mismatch problem is that the computation of sequence-level evaluation metrics such as BLEU, is not differentiable. Therefore, it cannot be optimized by gradient descent and backpropagation.

What is BLEU score? This metric depends on a few parameters:

- N-grams (usually N=4)
- Case sensitivity
- Brevity, denoted by ρ. It is used for penalizing short translation:

$$P(i) = \frac{Matched(i)}{H(i)} \qquad BLEU_a = \{\prod_{i=1}^{N} P(i)\}^{1/N} \qquad \rho = \exp\{\min(0, \frac{n-L}{n})\}$$

H(i) is the number of i-grams in the hypothesis, *P* is precision, *n* is the length of the hypothesis, and *L* is the reference length.

Addressing Objective Mismatch Cont'd

To address the objective mismatch issue:

- Current solutions are based on REINFORCE (1992).
  - A class of reinforcement learning algorithms that can optimize any metric of interest.
  - Maximizing the expected *reward* of model samples and trains on sampled sequences by using Policy gradients.
  - Self-Critical Sequence Training (SCST) is a type of REINFORCE that uses CIDEr as the reward signal and utilizes the greedy output of the model as the baseline to reduce variance.

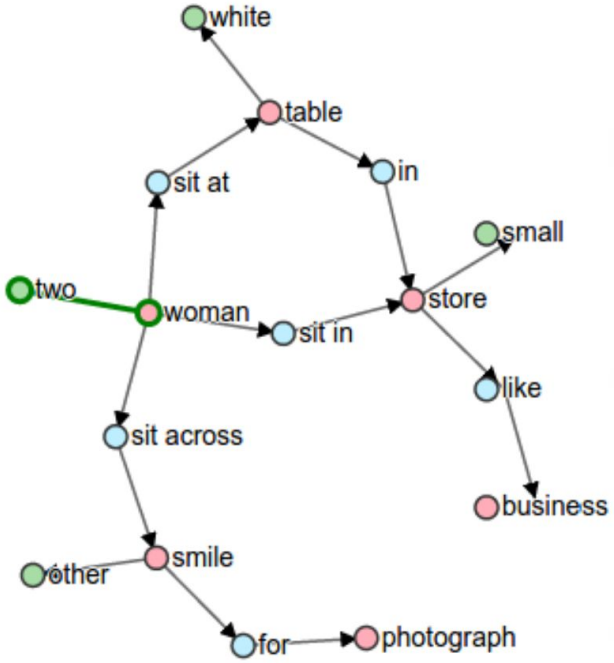
- What is CIDEr score? Consensus-based Image Description Evaluation
  - Also based on n-grams
  - Uses TF-IDF for weighting the importance of n-grams
  - measures the similarity of a generated sentence against a set of ground truth sentences written by humans.

Addressing Objective Mismatch Cont'd

- SPICE
  - Most recent metric proposed in 2016.
  - Compares generated sentences and reference sentence from a semantic similarity perspective by parsing them into a scene graph.
  - It outperforms CIDEr in terms of capturing human judgements.
  - In a recent work in 2017, it is shown that CIDEr and SPICE can be combined together to generate better captions.

"Two women are sitting at a white table"

"Two women sit at a table in a small store"



Datasets

- Early Datasets are from specific domains such as cooking and movie.
  - TACos (2013), YouCook (2013).
  - Their vocabularies and the amount of data are very limited.

- Newer datasets:
  - MPII-MD (2015) and M-VAD (2015) are created from audio description for movies.
  - Number of clips and vocabulary size are larger.

- Open domain datasets:
  - MSVD (2011) and contain web videos from different categories, but it has a limited size.
  - MSR-VTT (2016) is the first large-scale dataset. It contains 10,000 clips from 20 categories such as music, sports, and movies. It has the most descriptions and largest vocabulary size.

Evaluation

Below are the evaluation of some of some popular models on the MSR-VTT dataset. Three scores are reported: BLEU, METEOR, and CIDEr.

Method	T	B@4	M	C
MMVD [Ramanishka *et al.*, 2016]	M	40.7	28.6	46.5
Attention Fusion [Hori *et al.*, 2017]	M	39.7	25.5	40.0
MA-LSTM [Xu *et al.*, 2017]	M	36.5	26.5	41.0
HACA [Wang *et al.*, 2018c]	M	43.4	**29.5**	49.7
Temporal Att. [Yao *et al.*, 2015]	A	34.8	25.1	36.7
hLSTMat [Song *et al.*, 2017]	A	38.3	26.3	-
MGSA [Chen and Jiang, 2019a]	A	**45.4**	28.6	50.1
LSTM-E [Pan *et al.*, 2016]	S	36.1	25.8	38.5
M&M TGM [Chen *et al.*, 2017]	S	44.3	29.4	49.3
RL Ent [Pasunuru and Bansal, 2017]	R	40.5	28.4	**51.7**

In terms of CIDEr score, the RL Ent model outperforms other by a clear margin, particularly because it is directly optimized for this metric.

Conclusion and Future Work

Conclusion:

- We reviewed recent deep learning video captioning methods.
- Computer Vision and Natural Language Processing are highly used together.
- We also learned about metrics and datasets used for video captioning.

Future Directions:

- Modeling Object Interaction
  - In complex videos there are multiple interaction and visual relationships between the objects. They cannot be easily captured by spatial and temporal attention methods.
- Improving Event Proposal
  - leverage finer-grained information (such as visual concepts) for producing event proposals.
- Novel Decoder Structures
  - LSTMS are very popular. However, it is not easy to understand how much visual information has contributed to the generation of a certain word.

References

1. Chen, Shaoxiang, Ting Yao, and Yu-Gang Jiang. "Deep learning for video captioning: a review." Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press, 2019.
2. Anderson, Peter, et al. "Spice: Semantic propositional image caption evaluation." European Conference on Computer Vision. Springer, Cham, 2016.
3. http://ssli.ee.washington.edu/~mhwang/pub/loan/bleu.pdf
4. https://en.wikipedia.org/wiki/Mel-frequency_cepstrum