Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning

Madisetty Uday Theja (CS14B044)

Guide: Dr. C. Chandra Sekhar

Overview

What is Video Captioning ?

Video captioning is the task of automatically annotating videos with natural language descriptions.

A person pours milk into a bowl of rice and stirs it with a wooden spoon

Encoder - Decoder Framework

- **Encoder:**
  We use a convolutional neural network (CNN) encoder to extract compact feature vectors of each frame in the video which contains the relevant visual information.
- **Decoder:**
  The output of the CNN encoder is fed into the decoder which then tries to decode the visual information into a natural language output.

Basic LSTM for Video Captioning

- **Feature extraction using CNN encoder**:

$$V = \{v_1, v_2, \ldots, v_n\} = \varnothing_E(z) \tag{1}$$

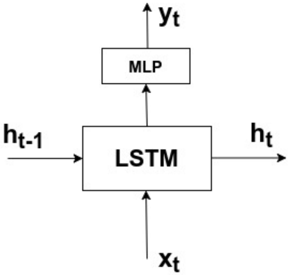
where $\varnothing_E$ is the encoder, z is the input video, v_i is the feature vector of i^{th} frame and n is the number of frames.

- **Initial LSTM state computation**:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} v_i \tag{2}$$

$$h_0, s_0 = [W^{ih}; W^{ic}]\mu \tag{3}$$

where W^{ih} and W^{ic} are parameters to be learned, h_0 and s_0 are initial hidden state and cell state of LSTM.



- **LSTM update**:

$$h_t, s_t = LSTM(x_t, h_{t-1}, s_{t-1}) \qquad (4)$$

where h_t and s_t are hidden state and cell state of LSTM and x_t is the word embedding at time step t.

- **Probability distribution computation**:

$$p_t = softmax(U_p \phi(W_p[h_t] + b_p) + d) \qquad (5)$$

where h_t is the hidden state of LSTM, U_p, W_p, b_p and d are parameters to be learned

Temporal Attention

- The feature representation of the video V and the bottom LSTM layer's hidden state h_t are fed through a single layer neural network. Then, *softmax* function is used to compute the **attention weights** over n frames, at each time step t.

$$\epsilon_t = w^T \tanh(W_a h_t + U_a V + b_a) \tag{6}$$

$$\alpha_t = softmax(\epsilon_t) \tag{7}$$

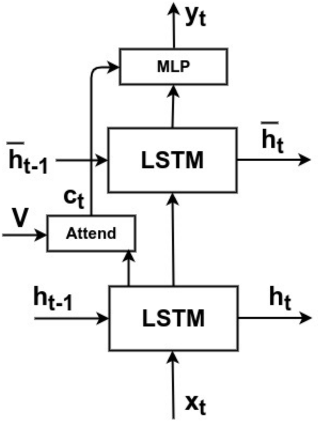
where w^T, W_a, U_a and b_a are the parameters to be learned.

- $\alpha_t \in \mathbb{R}^n$ gives the relevance of each frame at a time step t.
- The attention weights α_t^i learned through attention using Equation 7 are used compute the **context vector** c_t by taking the **dynamic weighted sum** of the video features at each time step t:

$$c_t = \frac{1}{n} \sum_{i=1}^{n} \alpha_t^i v_i \tag{8}$$

where the number of frames in the video is denoted by n.

Hierarchical LSTM with Temporal Attention for Video Captioning (hLSTMt)



V : feature representation of video
x_t : word embedding
h_t : hidden state of bottom LSTM
$\bar{h}_t$: hidden state of top LSTM
c_t : context vector
y_t : output word

Visual and Non-Visual Words

- **Visual words** :
  - Require visual information for prediction.
  - eg., "gun", "shooting"
- **Non-visual words** :
  - Do not require visual information for prediction.
  - Can be predicted by using language context information.
  - eg., "a", "of", "the", "sign" (after "behind a red stop"), "phone" (following "talking on a cell").
- Imposing attention mechanism on the non-visual words can mislead and decrease the performance of video captioning.

Adjusted Temporal Attention

- The hidden state of the bottom LSTM layer h_t is used to compute the adjusted gate β_t:

$$\beta_t = sigmoid(W_s h_t) \tag{9}$$

  where W_s is the parameter to be learned.
- β_t is the adjusted gate which is projected to the range $[0, 1]$.

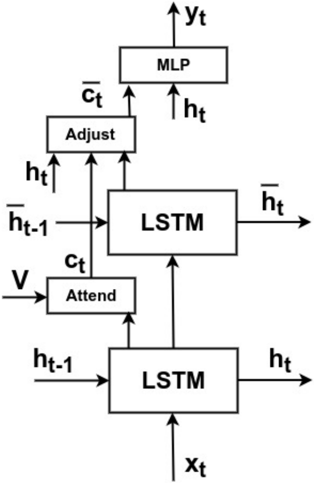
Final Context Vector

- The hidden state of the top LSTM layer $\bar{h}_t$, context vector c_t and adjusted gate β_t are used to compute the final context vector $\bar{c}_t$:

$$\bar{c}_t = \beta_t c_t + (1 - \beta_t)\bar{h}_t \tag{10}$$

- $\beta_t = 1$: complete visual information is considered using the context vector c_t for predicting the next word.
- $\beta_t = 0$: no visual information is considered and the next word is predicted using only the language context model.

Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning (hLSTMat)



V : feature representation of video

x_t : word embedding

h_t : hidden state of bottom LSTM

$\bar{h}_t$: hidden state of top LSTM

c_t : context vector

$\bar{c}_t$: final context vector

y_t : output word

Mean Aggregation Models

We try to reduce the number of parameters in the model to learn the attention weights better.

- **hLSTMat-PCA**
  - To reduce the dimensionality of the video feature representation, Principal component analysis (PCA) is used.
  - The dimension of the mean of the video features of each video is reduced to the LSTM size using PCA.
  - This is used as the new video representation V in Equation 6.

- **hLSTMat-KMeans**
  - To reduce the dimensionality of the video feature representation, the output of the CNN is clustered using **K-Means algorithm** with 3 **k-centers**.
  - The video V in Equation 6, is now represented as **(k-centers, mean at each k-center)**.
  - The weights for the k-centers are then passed through a single layer neural network to get the weights for each frame of the video.

The Microsoft Video Description Corpus (MSVD)

This dataset contains 1970 short videos with various natural language annotations for every video. It has approximately 80000 human annotated video-description pairs. The dataset is split into validation, test and training set with 100, 670 and 1200 videos respectively.

Table: Dataset Description

Name	#Total	#Training	#Validation	#Testing	#Vocabulary
MSVD	1970	1200	100	670	9433

The Effect of Different CNN Encoders

Table: Effect of different CNN encoders

Model	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR
VGG19	81.2	70.1	61.0	51.1	31.7
ResNet-50	83.1	73.0	64.2	54.6	32.9
ResNet-152	83.0	73.5	65.1	55.3	33.7
Inception-v3	83.8	74.0	65.3	**55.7**	**33.6**

- Inception-v3 performs the best with BLEU@4 score of 55.7% and METEOR score of 33.6%.
- ResNet-152 is very close in performance to Inception-v3.
- ResNet-152 seems to perform better than ResNet-50.

Architecture Exploration and Comparison

This experiment explores the architecture and studies the influence of the attention mechanisms. The experiments are conducted on MSVD dataset and use Inception-v3 as the CNN encoder.

Table: Effect of different types of the decoder

Decoder	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR
Basic LSTM	79.6	69.7	60.3	49.8	32.1
hLSTMt	83.0	72.8	63.2	53.6	32.9
hLSTMat	83.8	74.0	65.3	**55.7**	**33.6**

Results

ground truth : a soccer player making a long goal
basic LSTM : two teams are playing
hLSTMt : a man is running
hLSTMat : a soccer player makes a goal

ground truth : a jackal is walking around in a field
basic LSTM : a dog is catching a fish
hLSTMt : a dog is walking
hLSTMat : a animal is walking through a field

ground truth : a woman is putting on gold eyeshadow
basic LSTM : a woman is putting a stick in her mouth
hLSTMt : a girl is applying makeup
hLSTMat : a woman is applying eye makeup

Temporal Attention Analysis

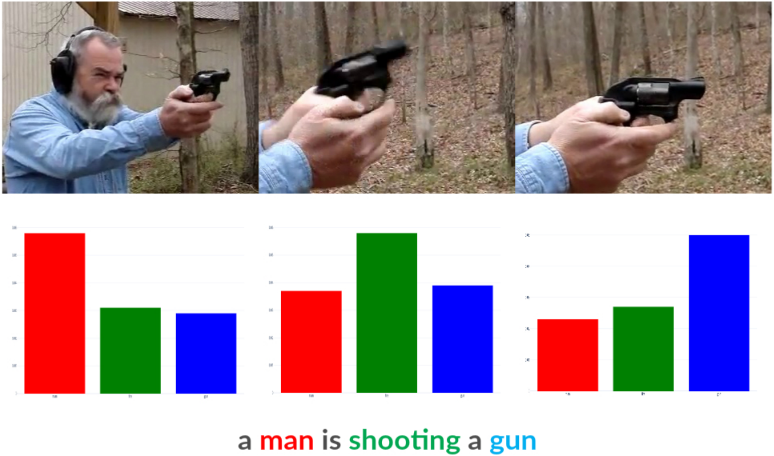
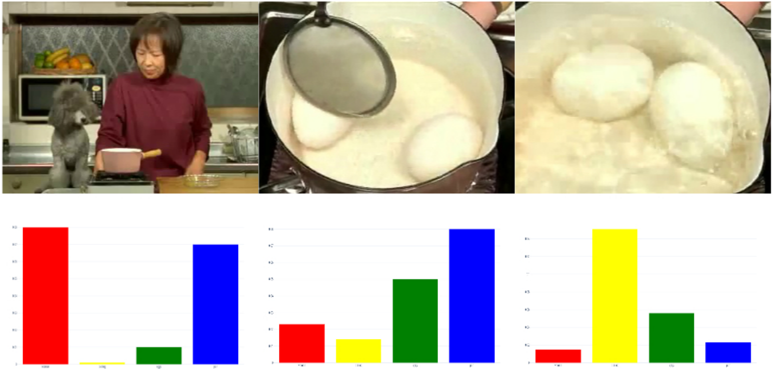


Figure: Video of a man shooting a gun



a woman is boiling eggs in a pan

Figure: Video of a woman boiling eggs in a pan

Adjusted Temporal Attention Analysis

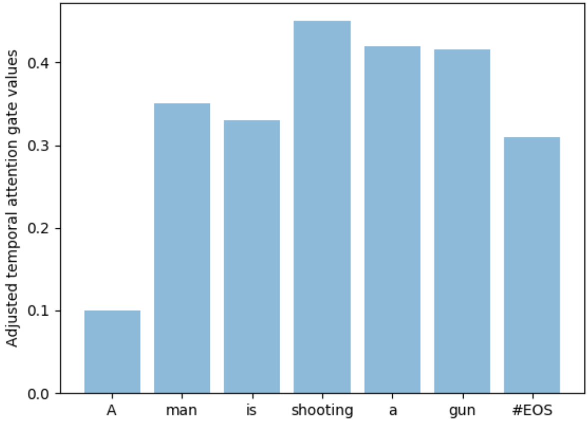


Figure: Video of a man shooting a gun

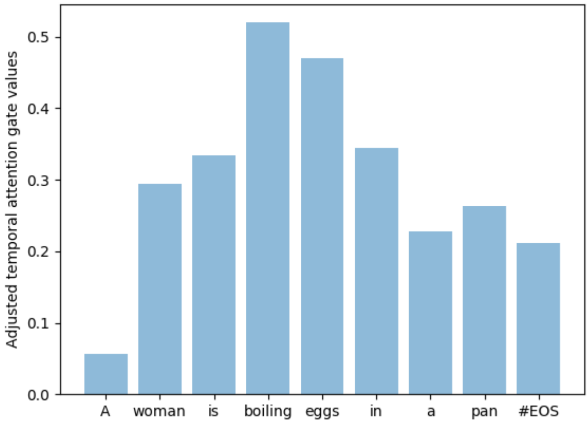


Figure: Video of a woman boiling eggs in a pan

Mean Aggregation Models Analysis

This experiment compares the performance of the mean aggregation models. The experiments are conducted on MSVD dataset and use Inception-v3 as the CNN encoder.

Table: Comparison of mean aggregation models

Decoder	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR
hLSTMat-PCA	81.5	71.0	61.9	51.3	33.7
hLSTMat-KMeans	82.9	72.4	63.4	53.3	34.1
hLSTMat	83.8	74.0	65.3	55.7	33.6

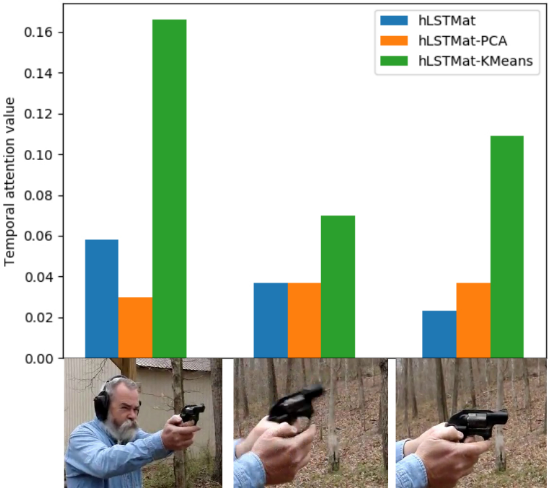


Figure: Temporal attention values while generating the word "man" in the caption

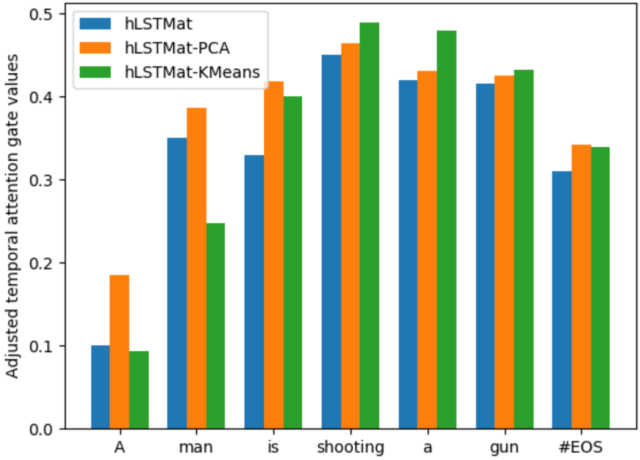


Figure: Adjusted temporal attention plots of a video in MSVD dataset comparing hLSTMat, hLSTMat-PCA and hLSTMat-KMeans as decoder

Conclusion and Future Work

- Experiments show that temporal attention and adjusted temporal attention improve the performance of the model and achieve state-of-the-art performance on MSVD dataset.
- The attention weights must further be fine tuned in the model
- Currently, the model uses only spatial visual information. Incorporating the model with both **temporal and spatial visual information** could further improve the performance of the model.

References

Jingkuan Song, Lianli Gao, Zhao Guo, Wu Liu, Dongxiang Zhang and Heng Tao Shen (2017)

Hierarchical LSTM with Adjusted Temporal Attention for Video Captioning

Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pages 2737–2743.

Questions ?