CPSC 8430
Deep Learning Homework-2
Avinash Komatineni

Github Link: https://github.com/avinash-komatineni/DeepLearning_HW2

Introduction:

In this video caption generation, we will take input as a short video, and we will generate output as captions that describes the video. Here we will use sequence to sequence model to train and test to implement video caption generation.

Dataset:

We have taken MSVD Dataset, it consists of 1550 video snippets, each with a duration of 10 to 25 seconds. For the purpose of our project, we split the dataset into two subsets - one with 1450 videos for training and another with 100 videos for testing.

Requirements:

Python
Torch
MVSD Dataset
Scipy

Data preprocess:

To initiate our project, we begin by loading the label file and constructing a dictionary that includes all the key words from the captions of the videos. We then examine how frequently each word occurs in a video's caption, and we assign a unique index to each word and its reverse process for all caption data. This allows us to determine the significance of each word for a particular video. Additionally, there are some tokens to consider: <PAD>, <BOS>, <EOS>, and <UNK>. The <PAD> token is used to ensure sentences are of the same length, while <BOS> indicates the beginning of a sentence in generating an output. <EOS> signifies the end of an output sentence, and <UNK> is used when a word is not in the dictionary, or it can be ignored altogether.

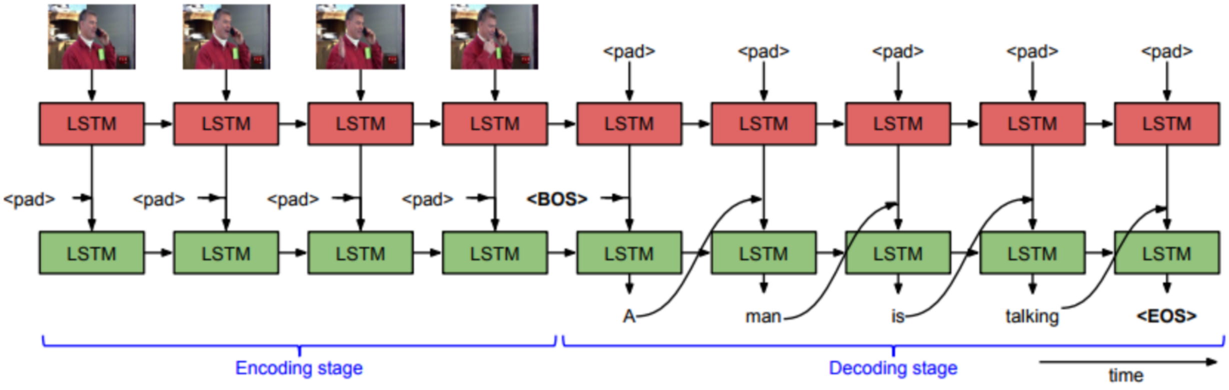
Sequence to sequence Model:

The Sequence to sequence model consists of two main components: an encoder and a decoder. The encoder is responsible for processing the input video frames and producing a fixed-length representation of the video. This is achieved by an RNN (GRU) to encode these features into a fixed-length vector.

The decoder takes the encoded video representation and generates a sequence of words that describe the video. It does this by using another RNN (GRU) to decode the fixed-length vector into a sequence of words. During training, the S2VT model is optimized to generate the correct sequence of words that accurately describes the input video frames.

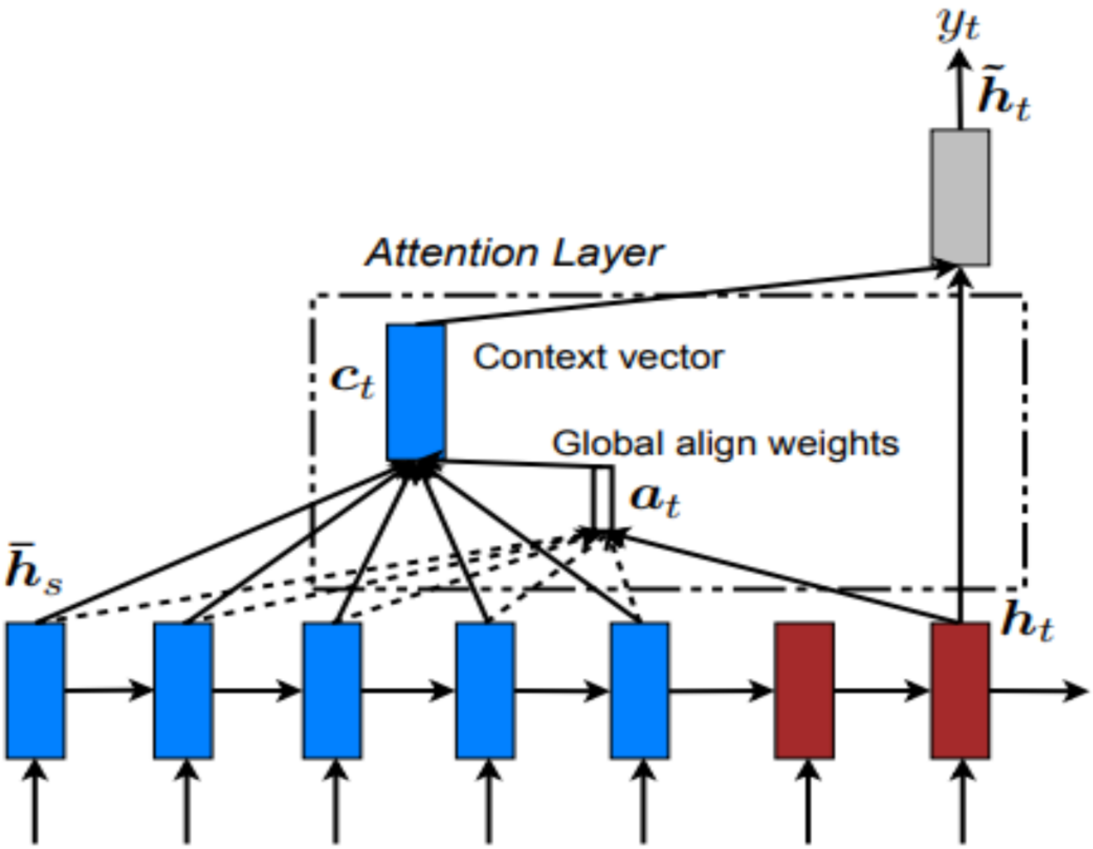
Approach:

In the next step, this model utilizes two layers of Gated Recurrent Units (GRUs) as its RNNs. In the first layer, the video is processed and encoded using the encoder network. The decoder network generates an output based on the encoded video. During the decoding process of the Sequence to sequence model, the captions are segmented into beginning and ending verses using tokens. The model then processes the encoded video and generates the actual words corresponding to the video.



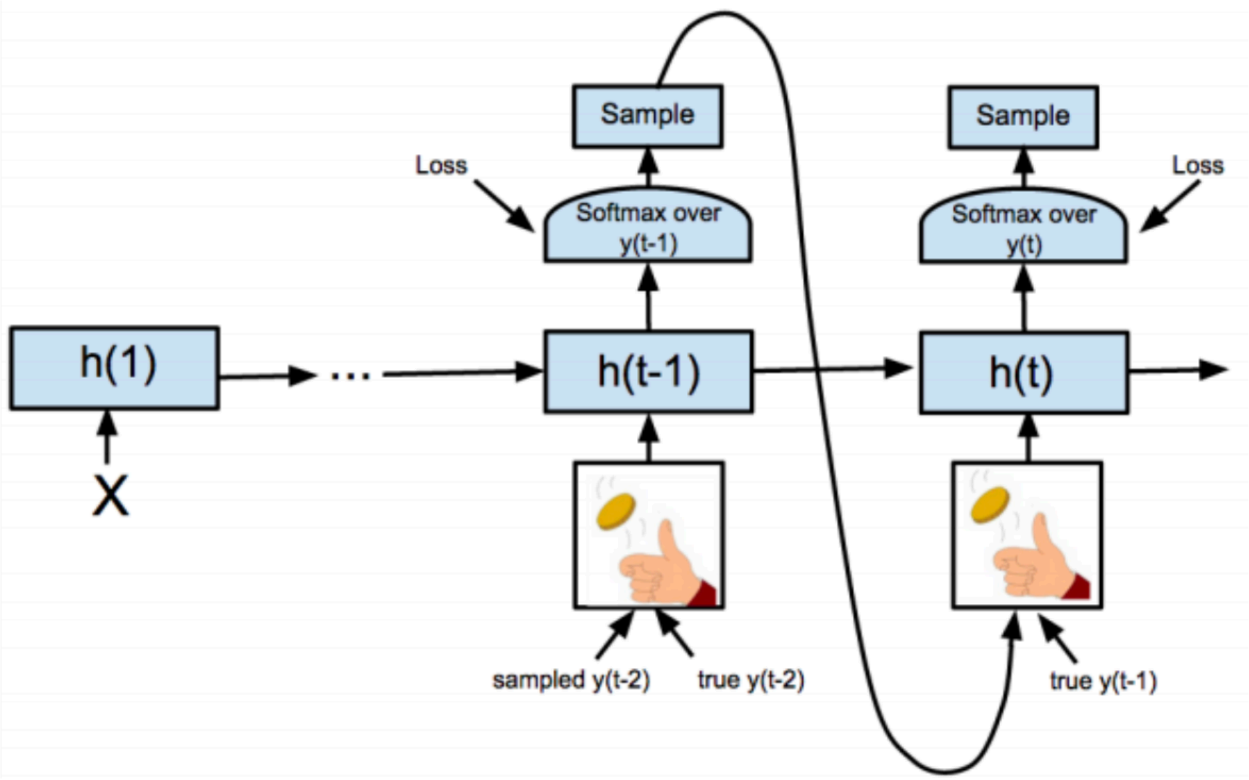
Attention layer:

an attention layer is added to the encoder hidden states to improve the quality of the generated captions. The attention mechanism helps the model focus on different parts of the encoded video when generating each word in the caption. During the decoding process, the attention layer calculates a weight for each frame of the video based on how relevant it is to the current decoding step. These weights are then used to calculate a weighted sum of the encoded video frames, which is passed as input to the decoder network along with the previous output word.



Schedule Sampling:

Schedule Sampling is a technique used in sequence-to-sequence models that allows the model to gradually transition from using the ground truth to using its own generated output as the input for the next time step during training.



As we trained a neural network model using the sequence-to-sequence with video to text architecture for video captioning. The model was trained for 75 and 100 epochs on a training dataset of size 1450, using a learning rate of 0.001 and a batch size of 128. The model had hidden layers with a size of 512 and used the Adam optimizer with a dropout rate of 0.3.

During training, we used a teacher learning ratio of 0.7, which helped to improve the stability and accuracy of the model. The vocabulary size was set to n>3, ensuring that the model could generate captions with a wide range of words and phrases.

After training, we evaluated the model on a test dataset of size 100 and achieved promising results in terms of both quantitative metrics and visual quality of the generated captions. These results demonstrate the effectiveness of the sequence-to-sequence architecture and the chosen hyperparameters in capturing the complex relationships between video frames and their corresponding textual descriptions. After evaluating the average bleu score for 75 epochs is 0.67646 and the average bleu score for 100 epochs is 0.69048.