CPSC 8430: Deep Learning
Homework 2 Report

Submitted by: Shwetha Sivakumar
CUID: C20140199

Video caption generation is the task of generating textual descriptions relevant to a given video, and it's well addressed by a sequence-to-sequence model. With its encoder-decoder mechanism, this model certainly fits to capture the intrinsic difference in length between input video frames and the length of its caption.

Several training techniques have been used in this work to optimize the model for performance: Attention, Scheduled Sampling, and Beam Search. Attention mechanism again serves to let the model concentrate on the most relevant parts of the video and discard those frames that are totally irrelevant or less important. This allows the model to generate more accurate and meaningful captions. Scheduled Sampling encourages the model to avoid overfitting by gradually allowing it to make its own predictions during training rather than being dependent altogether on the ground truth. This, therefore, helps in smoothing the transition, making the model robust, and thereby generalizing well on the test set.

This captioning process is further enhanced by Beam Search, allowing efficient traversal through possible sequences of captions. While the greedy decoding assumes the word having a maximum probability at a time step and computes just one sequence of words, Beam Search explores several potential sequences (beams) at each time step, keeping only the top-scoring ones at every step. This provides for a much more comprehensive search of possible captions while reducing computational load.

These combine to give higher-quality captions, attention, scheduled sampling, and beam search by optimizing the model's learning process and its accuracy, with the generated caption being more contextually appropriate and coherent.

Dataset Details

The video captioning dataset is made up of 1,450 training videos and 100 test videos, providing a broad basis on which to train and test the models for video captioning. Each video in the training set will have multiple captions, enabling the model to learn from the different ways of describing the same content. By contrast, the test set includes those videos for which the model would have to generate their captions by itself without access to ground-truth captions at evaluation.

The metadata regarding the videos, including the video IDs and captions for those, are stored in JSON files for training and test sets. To make life easier for the model, pre-extracted video features are provided in separate folders for training and testing from which the model could draw input. These precomputed features lighten the computational load and enable the model to focus on generating the captions. ID.txt files support the correct mapping of video features to their IDs by listing the video IDs. This helps in seamless integration into the captioning pipeline.

The dataset is a structured skeleton containing videos, captions, metadata, and precomputed features that make the training, testing, and evaluation of models for video caption generation easy.

For the benchmark, a sequence of steps regarding the baseline was conducted as follows:

Data Preprocessing:

The process started by loading the JSON file containing the captions. Counting the frequency of each word in the captions was followed by mapping the word to its index, and vice-versa. Special tokens were included like those of <PAD>, <SOS>, <EOS>, and <UNK>. Once the mappings had been made, the words in the captions were replaced with their corresponding indices. Then comes Caption Preprocessing and Generation of Annotated Caption Data. Once the words were replaced with their indices, the generation of the annotated caption data involved preprocessing the captions. In this data structure, video features could be associated with their respective captions without much hassle or complications, hence easing the training process. It's followed by feature data integration where the feature data from training files has been loaded, and a dictionary has been created to map each ID of a video with its respective features. Similarly, feature data loading from the test files resulted in a list of tuples, each tuple containing a video ID with its corresponding feature data. The final step was batch formation is to group video features with their corresponding caption indices. Sequences were padded to ensure consistency according to the maximum length of the captions so that each input in the model had a fixed size. This order was followed in the construction of the required data structure, hence preparing the captions and video features for fast processing during model training and evaluation. Such steps assuredly prepared the model with inputs that were well organized-a condition normally necessary for learning meaningful patterns in video caption generation.

Code link: https://github.com/shwethasivakumar/Deep-Learning/blob/main/hw2/hw2_1/pretrain.py

BLEU Score Calculation:

BLEU score calculation becomes an important step after data preprocessing for comparing the generated captions with reference captions. BLEU score calculation requires two measures: precision and brevity penalty.

Precision is measured based on how many n-grams of generated captions exactly match the reference captions' n-grams. In this regard, it can be normally done for a different number of values for n, such as unigrams, bigrams, trigrams, and so on. The obtained precisions for each n-gram are then averaged into a precision score.

Brevity penalty penalizes those generated captions which are much shorter than the reference captions. If the length of the generated caption is less than the reference, then the brevity penalty will be applied; it tries to discourage overly brief outputs, which ensures that unnaturally shortly generated captions are not achieved. The BLEU score usually is the geometric mean of precision scores over different n-grams, mostly for unigrams, bigrams, and trigrams, and sometimes higher. Then, it is multiplied by the brevity penalty.

First, for every video, the BLEU score of its candidate caption is calculated against its corresponding reference captions. Then, an average BLEU score across all videos is computed. The average BLEU score gives the general performance of how well the model is generating captions compared to the ground truth references and therefore gives a view of the model's performance in terms of accuracy and linguistic quality.
Code link: https://github.com/shwethasivakumar/Deep-Learning/blob/main/hw2/hw2_1/bleu_eval.py

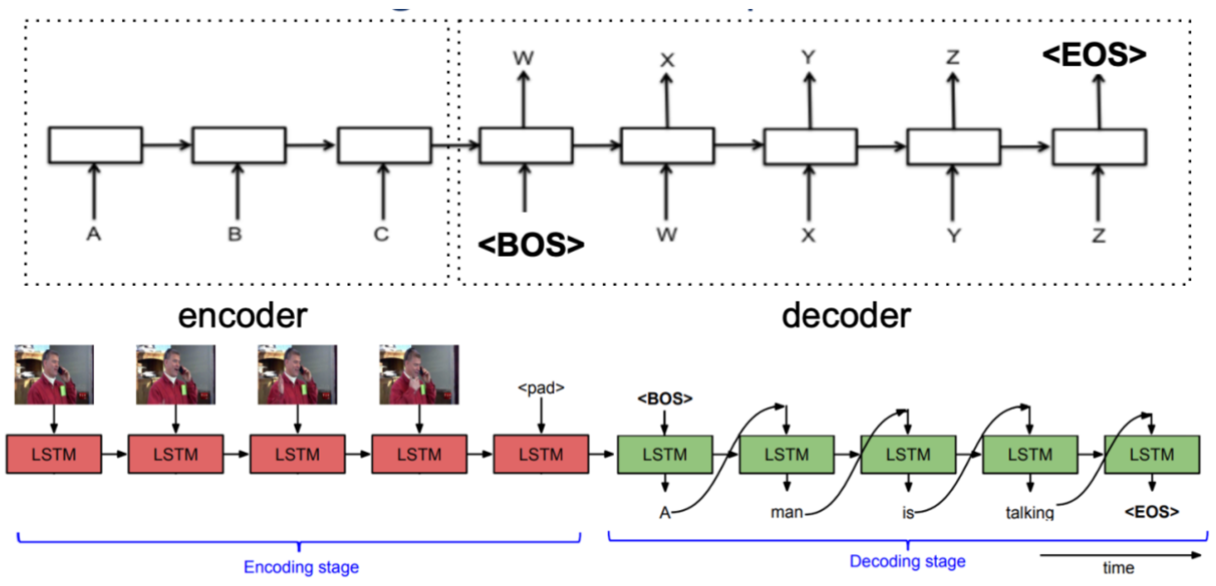
Data Training:

During the training process, pre-processed data in the form of captions and their respective video features are fed into the model. In the sequence-to-sequence architecture, it contains both an encoder and a decoder, each with an attention mechanism. The encoder compresses the input video features that pass through a GRU layer, capturing sequential dependencies of the features. The decoder performs caption generation in a word-by-word fashion while the attention mechanism aids the decoder by selecting relevant frames from the video, computing attention weights from the hidden state of the decoder and outputs from the encoder.

The model is trained in batches, running each batch for 50 epochs. It computes, at each step, the cross-entropy loss: this measures the discrepancy between the generated captions and the reference captions. Backpropagation tries to minimize the loss by updating model parameters. Finally, to optimize the training process, the Adam optimizer uses a learning rate set at 0.0001 to ensure efficient convergence.

The saving model-after the training process is completed-allows one to use this model in future evaluations or inferences, or to fine-tune it. In this way, it will ensure the capabilities of the model in making captions for unseen videos regarding what it had learned about representations of video features and corresponding text.

Code link: **https://github.com/shwethasivakumar/Deep-Learning/blob/main/hw2/hw2_1/train.py**



Testing and Caption Generation:

After training, it is used to generate captions for test videos. Model captions are compared against the ground truth captions. BLEU scores are calculated on the generated captions to verify the quality of the captions generated. It is a measure of the precision in the correctness of the model. Then, the average BLEU score is calculated, which becomes the model's score.

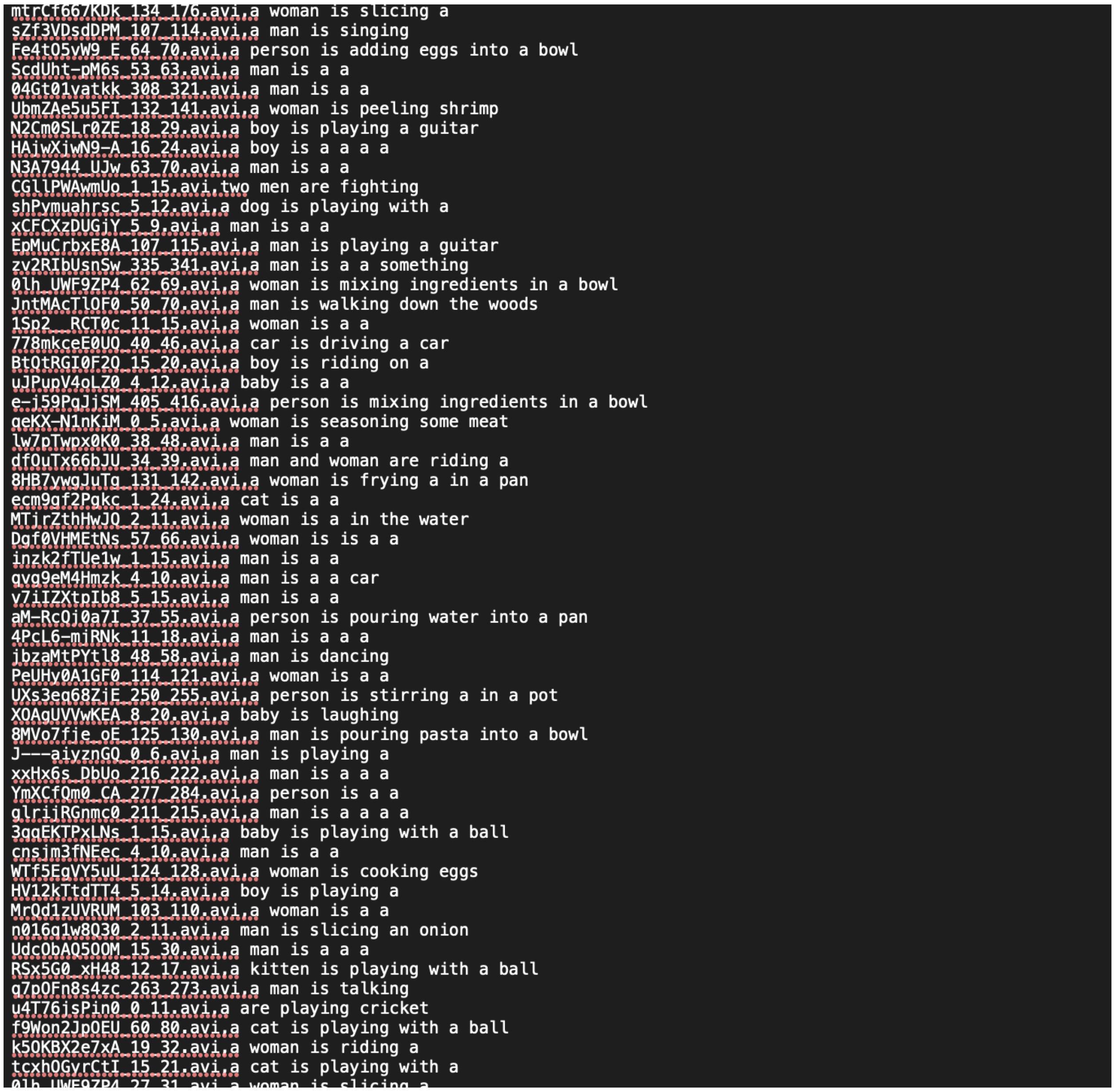
Besides, an output file is generated which contains the captions proposed by the model for each test video. It is the result of the generated caption. After that, save the trained model for further re-use, if necessary, to be used either for inference or re-fine tuning.

Code link: **https://github.com/shwethasivakumar/Deep-Learning/blob/main/hw2/hw2_1/caption_generation.py**

The below command is used to run the "caption_generation.py" script and provide with necessary arguments for generating and saving captions in an output file named "output11.txt":

```
python3 caption_generation.py "hw2/hw2_1/MLDS_hw2_1_data/testing_data/feat"
"hw2/hw2_1/output11.txt"
```

Output File Contents:



```
mtrCf667KDk_134_176.avi,a woman is slicing a
sZf3VDsdDPM_107_114.avi,a man is singing
Fe4tO5vW9_E_64_70.avi,a person is adding eggs into a bowl
ScdUht-pM6s_53_63.avi,a man is a a
04Gt01vatkk_308_321.avi,a man is a a
UbmZAe5u5FI_132_141.avi,a woman is peeling shrimp
N2Cm0SLr0ZE_18_29.avi,a boy is playing a guitar
HAjwXjwN9-A_16_24.avi,a boy is a a a a
N3A7944_UJw_63_70.avi,a man is a a
CGllPWAwmUo_1_15.avi,two men are fighting
shPymuahrsc_5_12.avi,a dog is playing with a
xCFCXzDUGjY_5_9.avi,a man is a a
EpMuCrbxE8A_107_115.avi,a man is playing a guitar
zv2RIbUsnSw_335_341.avi,a man is a a something
0lh_UWF9ZP4_62_69.avi,a woman is mixing ingredients in a bowl
JntMAcTlOF0_50_70.avi,a man is walking down the woods
1Sp2__RCT0c_11_15.avi,a woman is a a
778mkceE0UQ_40_46.avi,a car is driving a car
BtQtRGI0F2Q_15_20.avi,a boy is riding on a
uJPupV4oLZ0_4_12.avi,a baby is a a
e-j59PqJjSM_405_416.avi,a person is mixing ingredients in a bowl
qeKX-N1nKiM_0_5.avi,a woman is seasoning some meat
lw7pTwpx0K0_38_48.avi,a man is a a
dfOuTx66bJU_34_39.avi,a man and woman are riding a
8HB7ywgJuTg_131_142.avi,a woman is frying a in a pan
ecm9gf2Pgkc_1_24.avi,a cat is a a
MTjrZthHwJQ_2_11.avi,a woman is a in the water
Dgf0VHMEtNs_57_66.avi,a woman is is a a
inzk2fTUe1w_1_15.avi,a man is a a
qvg9eM4Hmzk_4_10.avi,a man is a a car
v7iIZXtpIb8_5_15.avi,a man is a a
aM-RcQj0a7I_37_55.avi,a person is pouring water into a pan
4PcL6-mjRNk_11_18.avi,a man is a a a
jbzaMtPYtl8_48_58.avi,a man is dancing
PeUHy0A1GF0_114_121.avi,a woman is a a
UXs3eq68ZjE_250_255.avi,a person is stirring a in a pot
XOAgUVVwKEA_8_20.avi,a baby is laughing
8MVo7fje_oE_125_130.avi,a man is pouring pasta into a bowl
J---aiyznGQ_0_6.avi,a man is playing a
xxHx6s_DbUo_216_222.avi,a man is a a a
YmXCfQm0_CA_277_284.avi,a person is a a
glrijRGnmc0_211_215.avi,a man is a a a a
3qqEKTPxLNs_1_15.avi,a baby is playing with a ball
cnsjm3fNEec_4_10.avi,a man is a a
WTf5EgVY5uU_124_128.avi,a woman is cooking eggs
HV12kTtdTT4_5_14.avi,a boy is playing a
MrQd1zUVRUM_103_110.avi,a woman is a a
n016q1w8Q30_2_11.avi,a man is slicing an onion
UdcObAQ5OOM_15_30.avi,a man is a a a
RSx5G0_xH48_12_17.avi,a kitten is playing with a ball
q7pOFn8s4zc_263_273.avi,a man is talking
u4T76jsPin0_0_11.avi,a are playing cricket
f9Won2JpOEU_60_80.avi,a cat is playing with a ball
k5OKBX2e7xA_19_32.avi,a woman is riding a
tcxhOGyrCtI_15_21.avi,a cat is playing with a
0lh_UWF9ZP4_27_31.avi,a woman is slicing a
```

Code link: https://github.com/shwethasivakumar/Deep-Learning/blob/main/hw2/hw2_1/output11.txt

Results:

The Average Bleu score = 0.611.