CPSC 8430
Fall 2024
Homework 2
Video Caption Generation using Seq2Seq with Attention
Abdullah Al Mamun

GitHub repository: https://github.com/abdullm/hw2
Best Model Location:
https://drive.google.com/drive/folders/10cb3_ZqyMoHtTGjJTMAANfJWKHplNfLz?usp=sharing

1. Introduction and Objective

The task of this homework was to develop a sequence-to-sequence (Seq2Seq) model with attention mechanism to generate captions for videos. The model uses a Recurrent Neural Network (RNN) framework, where an encoder processes the input video features and a decoder generates captions describing the video. The objective of this task is to implement a video captioning model, using the Microsoft Research Video Description Corpus (MSVD) video features (1450 videos for training and 100 videos for testing) and their corresponding captions. The evaluation metric used is the BLEU score, which compares the predicted captions with the reference captions.

2. Model Architecture

2.1. Encoder:

- Processes video features using a Long Short-Term Memory (LSTM) model.
- Input size: 4096 (matching the dimensionality of the video features)
- Hidden size: 512
- The encoder extracts features from the input video and returns both the outputs and hidden states to the decoder.

2.2. Attention Mechanism:

- Allows the decoder to focus on different parts of the encoded video sequence while generating captions.
- Improves performance by selectively attending to relevant parts of the video features during each decoding step.

2.3. Decoder:

- Another LSTM model that uses the hidden states of the encoder and attention mechanism to generate captions.
- Equipped with an embedding layer to convert word indices to dense vectors.
- A fully connected output layer predicts the next word in the caption sequence.
- Incorporates attention weights to guide focus on the relevant parts of the video features.

2.4. Beam Search:

- Used during inference to generate captions by maintaining the most likely sequences.
- Keeps a fixed number of the most probable sequences at each step and expands them to explore possible outputs.

3. Training and Loss Function

- Cross-Entropy Loss: Used for training
- Teacher Forcing: Applied during training, where the true next word is provided as input to the decoder instead of the predicted word with a certain probability to speed up convergence.

4. Training Configuration

- Number of epochs: 200
- Learning rate: 0.001
- Batch size: 32
- Hidden size of LSTM: 512
- Beam width for decoding: 3
- Best BLEU Score Tracking:
  - The model tracks the best BLEU score achieved during training.
  - The model parameters are saved whenever a new best BLEU score is found.

5. Evaluation and BLEU Score

The Bilingual Evaluation Understudy (BLEU) score is used to evaluate the quality of generated captions by comparing them to reference captions. A higher BLEU score indicates a closer match between the predicted caption and the ground truth. For this task, the provided *bleu_eval.py* script has been used (to import the *BLEU* function) to perform model performance evaluations.

6. Dataset

Training Dataset: 1450 videos from the MSVD dataset (provided) were used as the training dataset.

Testing Dataset: 100 videos from the MSVD dataset (provided) were used as the testing dataset.

7. Results

Output Files:

- *best_decoder.pth*: Decoder model of the best Seq2Seq model in terms of providing maximum BLEU score on the test dataset.
- *best_encoder.pth*: Encoder model of the best Seq2Seq model in terms of providing maximum BLEU score on the test dataset.
- *sample_output_testset_AAM.txt*: Contains the model's predicted captions for the test videos.
- *Average loss and BLEU score with Epochs.txt*: Contains the training loss and BLEU scores over the epochs.

Evaluation Results:

- The model's performance was evaluated using the BLEU score on the test dataset.
- During training, the model's performance improved over time with decreasing loss and increasing BLEU score.
- The best BLEU score found was 0.7581.

8. Files and Their Functions

- run_seq2seq_AAM.py:
  - This is the main script that implements the Seq2Seq model with attention.
  - It trains the model, evaluates it, and generates captions for test videos using beam search.
- bleu_eval.py:
  - Implements BLEU score evaluation for comparing generated captions with reference captions.
  - Used during model evaluation to calculate the BLEU score for the predictions.
- hw2_seq2seq_AAM.sh:
  - A shell script to automate the running of the main Python script (run_seq2seq_AAM.py) with appropriate arguments (data directory and output).
- sample_output_testset_AAM.txt:

    - Stores the predicted captions generated by the model for the test videos.
- Average loss and BLEU score with Epochs.txt:
    - Contains the average loss and BLEU scores at each epoch during training. This file helps in tracking the model's performance over time.