Michael Elrod

Video Caption Generation Project Report
GitHub Repository Link

1 Introduction

This project implements a video caption generation system using deep learning techniques. The system accepts video features as input and generates textual descriptions (captions) for the video content. The project is organized into several Python scripts that handle data processing, model architecture, training, and evaluation.

2 Project Structure

The project is divided into four main Python scripts:

- `test.py`: Responsible for testing and evaluating the trained model.
- `train.py`: Handles data processing and model training.
- `bleu_eval.py`: Implements BLEU score calculation to assess caption quality.
- `model.py`: Defines the neural network architecture for caption generation.

3 Model Architecture

The caption generation model follows an encoder-decoder architecture with an attention mechanism:

3.1 Encoder

The encoder (`EncoderNet`) processes the input video features:

- Reduces input feature dimensions from 4096 to 512.
- Applies dropout for regularization.
- Uses an LSTM to process the sequence of frame features.

3.2 Decoder

The decoder (`DecoderNet`) generates captions based on the encoded video features:

- Utilizes an embedding layer for word representations.
- Implements an attention mechanism to focus on relevant parts of the video.
- Uses an LSTM to generate words sequentially.
- Applies teacher forcing during training with a dynamic ratio.

3.3 Attention Mechanism

The attention module (`Attention`) helps the decoder focus on important parts of the encoder output:

- Computes attention weights through linear transformations.
- Produces a context vector by weighting the encoder's output.

4 Training Process

The training process, implemented in `train.py`, includes the following steps:

- Data loading and preprocessing using a custom `DataProcessor` class.
- Dictionary creation for word-to-index mapping.
- Model initialization and training loop.
- Usage of Adam optimizer and CrossEntropyLoss.
- Saving the trained model and vocabulary.

5 Testing and Evaluation

The testing process, handled by `test.py`, includes:

- Loading the trained model and test data.
- Generating captions for test videos.
- Calculating BLEU scores using `bleu_eval.py`.
- Outputting results and the average BLEU score.

6 BLEU Score Calculation

The BLEU (Bilingual Evaluation Understudy) score is calculated in `bleu_eval.py`:

- Implements n-gram counting and clipping.
- Calculates the brevity penalty.
- Computes the final BLEU score using the geometric mean of precisions.

7 Results

The video caption generation system achieved promising results:

- Average BLEU Score: 0.6645

This score indicates a high level of similarity between the generated captions and reference captions, showing that the model can produce relevant and accurate descriptions of video content. A BLEU score of 0.6645 is considered good in caption generation tasks, highlighting the effectiveness of the architecture and training approach.

8 Conclusion

This project successfully implements a video caption generation system using deep learning techniques. The encoder-decoder architecture, combined with the attention mechanism, has proven effective in generating relevant captions based on video features. The achieved BLEU score of 0.6645 demonstrates the model's ability to generate high-quality captions that align well with human-generated references.