DEEP LEARNING HOMEWORK-2

VIDEO CAPTION GENERATION

VARSHITHA BONGULURU

https://github.com/Varshitha0/DeepLearningHomeWork-2.git

INTRODUCTION

It is difficult to create natural language subtitles for movies because computer vision and natural language processing must deal with the varied length inputs and outputs of videos. An encoder-decoder system learns video representations and produces sentences based on these representations by employing two LSTM units for encoding and one for decoding. The MSVD dataset is used to assess the model, while Beam search is used to generate captions. In contrast to conventional techniques, it makes use of a dynamic sequence-to-sequence model in which the decoder's attention layer focuses on relevant data to improve the output captions' accuracy and relevancy. Taking advantage of GPU acceleration using Python, Pandas, NumPy, Torch, Pickle, SciPy, and CUDA, this method offers an advanced solution for automatic video captioning that addresses the complex nature of real-time video footage.

One tool for creating captions for videos is the S2VT model. It creates captions by extracting features from video frames using a data pre-processor script. An attention layer is included in the model design to help with input element focus, along with encoder and decoder layers. To reduce the difference between generated and actual captions, the training script makes advantage of gradient descent and backpropagation. For test video inputs, the testing script creates captions and applies the BLEU score to analyse them. The testing script may be easily executed with the help of a shell script named Hw2_seq2seq.sh. An attention layer is another feature of the model's architecture that allows it to focus on a particular input component.

DATA PREPROCESSING

The data pre-processor script for video captioning reads training video features and captions, storing them in feature and caption dictionaries, respectively. It includes English captions, adding words to the dictionary if they appear more than three times, while less frequent words are marked as <UNK> for "unknown" to maintain a manageable dictionary size. The script employs four special tokens: <PAD> for sentence padding, <BOS> for marking the beginning of a sentence, <EOS> for the end of a sentence, and <UNK> for unknown words. This setup prepares the data for the model by converting sentences into sequences of numerical indices, facilitating uniform input processing and efficient model training.

With an average BLEU score of 0.709, the model demonstrated a comparatively high level of accuracy. Processing 10 of the 1450 batches took an average of 33.19 seconds. For features and testing labels, respectively,.npy and .json files were used in the setup. Three essential Python dictionaries were used in the experiment to manage the data:

Using word-dict to build vocabulary, uncommon terms are excluded.

W2i for word-to-number index mapping.

I2w for the index-to-word mapping in reverse.

These components made it easier to train and assess the model for producing and comprehending English.

MODEL

The S2VT model is a tool used to generate captions for videos. It uses a data pre-processor script to extract features from video frames and generate captions. The model architecture includes an encoder and decoder layers, with an attention layer for focusing on input elements. The training script uses backpropagation and gradient descent to minimize the difference between generated and actual captions. The testing script generates captions for test video inputs and evaluates them using the BLEU score. A shell script called Hw2_seq2seq.sh is provided for easy execution of the testing script. The model's architecture also includes an attention layer for focusing on specific input elements.

TESTING AND TRAINING

Here are the model parameters

Epochs: 100

Learning rate: 0.0001

Batch size: 10

Loss Function: nn.CrossEntropyLoss()

Optimizer: Adam

Training sample size: 1450

Video Features Dimension: 4096

Video Frame Dimension: 80

In the testing phase, the model generates captions for test videos, which are then compared to the actual ground truth captions to calculate BLEU scores which is 0.709. This measures the model's accuracy in producing captions that closely match human-generated text, offering a quantitative evaluation of its performance.