Image Captioning on the COCO Dataset

Trevor Powers, Nisarg Sureshkumar Patel, James McKee

Description:

Image captioning is the process by which a given ML model receives as input an image and outputs natural language describing what is observed in the image. This relatively new field of machine learning lends itself to a variety of applications, from virtual assistants to the development of tools for the disabled. Conceptually, image captioning requires two different types of knowledge: knowledge about what is occurring in the image, and sufficient knowledge of words to express what is occurring in the image. This is done by utilizing both computer vision and natural language processing techniques, where the image is encoded, and then decoded into a sequence of text. In this project, we explore a number of different methods of implementing image captioning on a COCO dataset, with emphasis on comparing different implementations of the decoder.

Dataset:

Common Objects in Context - COCO[1,2]

This dataset has been prepared by Microsoft for a host of different ML tasks including object segmentation, recognition in context, key points, and image captioning. For our purposes, the dataset includes 5 separate, sentence-length captions for each image. We intend to copy each image 5 times and associate each caption with one instance of that image. If this proves to be too unwieldy for our purposes, we will randomly downsample first.

Methodology and Expected Results:

For our methodology, we explore three different networks - a CNN-RNN, CNN-LSTM, and a CNN-Transformer. The CNN-RNN feeds the output of a CNN into an RNN, such that the CNN 'reveals' the relevant information from the image which is then interpreted by the RNN to generate a sentence. This will use the keras.layers.SimpleRNN method from Tensorflow to create a simple, fully connected RNN.

This architecture uses CNN for feature extraction on input data(images) which is combined with Long Short Term Memory(LSTM) to perform sequence prediction on feature vectors. It can be done using keras.layers.LSTM method.

For the final model, we use a CNN-Transformer. A transformer is a machine learning technique which makes use of attention, which weighs each part of the input data. Since its development by Google Brain in 2017, transformers have quickly become the state-of-the-art for NLP tasks, and we expect our CNN-Transformer model to result in the best overall performance.

[1] https://cocodataset.org/#home

[2] https://arxiv.org/pdf/1405.0312.pdf

The ablation strategy overall is trying three different network models after the CNN. If time and training permit, we will also experiment with different parameterizations for both the CNN and the decoders, potentially along with different dropout strategies.

Performance Evaluation (Same for all Networks):

The following methods are the common methods for evaluating image captioning. We will borrow Microsoft's implementation of these methods.[3]

1. BLEU
2. METEOR
3. ROUGE-L
4. CIDEr

Timeline:

-Week 9: Examine the dataset, Begin building models
-Week 10: Finish building models, Start training models
-Week 11: Finished training models, Begin model evaluation
-Week 12: Model evaluation, Begin report
-Week 13: Finalize code demonstration, Finalize report
-Week 14: Code demonstration, Report submission, Code submission

Responsibilities:

There are three members in our group. Since we are training three models, we will each implement a model: Trevor will implement the CNN-Transformer, Nisarg will implement the CNN-LSTM, and James will implement the CNN-RNN. We will then regroup and collectively analyze the outputs of the models using the NLP methods provided by Microsoft for its COCO captioning challenge. Visualizing the results and writing the report will also be fully collaborative.

[3] https://github.com/tylin/coco-caption