Mini-Project

Image Captioning

Introduction

As we know RNN's are very powerful, especially for sequential data modelling. There are basically four types of RNN's namely – one to one, one to many, many to one, many to many. Image captioning is an application of 'one to many' RNN's. For a given input image model predicts the caption based on the vocabulary of train data. We are given the Flickr8k dataset for this group project. Flickr8k dataset is small in size that means it can be trained easily on our lappies. We see that the data is properly labelled. We're provided 5 captions with each image, that's quite generous.

Understanding the Data

Data Pre-processing and cleaning plays a vital role in the whole model building process. We need to comprehend the data very well for building accurate models. We have two folders in the Flickr8k dataset. First is Flickr8k_Images which contains 8092 images in JPEG format with different shapes and sizes. Of which 6000 are used for training, 1000 for tests and 1000 for development. and another is Flickr8k_text which Contains text files describing train_set ,test_set. Flickr8k.token.txt contains 5 captions for each image i.e. total 40460 captions.

Exploratory Data Analysis: As we saw we have two types of data namely-Images, Texts (Captions). The size of the training vocabulary is 7371. The top 10 most frequent words are ('a', 46784), ('in', 14094), ('the', 13509),('on', 8007), ('is', 7196), ('and', 6678), ('dog', 6160), ('with', 5763), ('man', 5383) and ('of', 4974).

We assume that since the words which occur very less does not carry much information. So, we are considering words with a frequency of more than 10. So, we will get a count plot between values and counts which depicts distribution of word count. Now, we will plot between words and counts for top 50 words distribution. Then, we plot between words and counts that depicts the distribution of least frequent words with a frequency of less than 1. Then we plot between counts and average sequence length that depicts average caption length per image in the training. Then we calculate mean, standard deviation, percentile and found that maximum sequence length is 37.

Featurizing Images

Image Featurization allows us to make effective use of relatively small sets of labeled images that would not be sufficient to train a deep network from scratch. This is because we can re-use the lower-level features that the pre-trained model had learned for more general image classification tasks. Here, we will be using Inception because it is a low size file and it is faster to train. We will remove the soft-max layer from inception as we want to use it as a feature extractor. For a given input image, inception gives us a 2048 dimensional feature extracted vector. For every training image, we are resizing it to (299,299) and then passing it to Inception for feature extraction. Remember to save the train_image_extracted dictionary. it will save a lot of time if you are fine-tuning the model.

Caption Preprocessing

Each image in the dataset is provided with 5 captions. Captions are read from Flickr8k.token.txt file and stored in dictionary k:v where k = image id and value = [list of caption].

Since there are 5 captions for each image and we have preprocessed and encoded them in below format: “startseq “ + caption + “ endseq” The reason behind startseq and endseq is, startseq: Will act as our first word when feature extracted image vector is fed to decoder. It will kick-start the caption generation process. enseq: This will tell the decoder when to stop. We will stop predicting word as soon as endseq appears or we have predicted all words from train dictionary whichever comes first.

Sequential Data Preparation

For the understanding purpose, we will consider the above image and corresponding sequences where a bunch of people are swimming in water. First feed the image to inception and get feature extracted 2048 dimensional vectors.

Caption: startseq a bunch of people swimming in water endseq.

Then converting the sequence to numerical with the help of vocabulary.

There is a problem if we fit all the data points at once to model. Since we have a total of 40k captions. Max length of the caption is 37. So each caption is encoded into a sequence of 37. Let's assume on an average, to encode a sequence we need 10 rows. And, each word in a sequence will be embedded into a 300-dimensional glove vector. So considering 1 byte for each number , our final data matrix will occupy.. 40k x 10x((37 x 200) + 2048) ===> approx. 3.7GB at least.

Now for it's solution Rather than getting whole data at one time, use data_generator to generate data in batches.

BLEU

BLEU stands for Bilingual Evaluation Understudy.

It is an algorithm, which has been used for evaluating the quality of machine translated text. We can use BLEU to check the quality of our generated caption.

- BLEU is language independent
- Easy to understand
- It is easy to compute.
- It lies between [0,1]. Higher the score better the quality of caption

We calculate BLEU score as follows: For example: predicted caption= "the weather is good" references: the sky is clear, the weather is extremely good. Firstly, convert the predicted caption and references to unigram/bigrams.

Predicted: (the, weather), (weather, is), (is, good)

References: (the, sky), (sky, is), (is, clear), (the, weather), (weather, is), (is, extremely), (extremely, good)

BLEU(Bigram): 1/3(the, weather) + 1/3(weather, is) + 0/3(is, good) = 2/3 = 0.6666

BLEU tells how good is our predicted caption as compared to the provided 5 reference captions.

Inference/Conclusion

For a given feature extracted test image and startseq as i/p to model, we get a distribution of probability over all the words in the vocabulary. The word corresponding to the index of maximum probability is the predicted word. We stop predicting when the word "endseq" appears.

Image 1: man with blue shirt is sitting in the bench with his arms in the air
BLEU value: 1.6

Image 2: People are riding bicycle in street
BLEU value: 1.5

Image 3: People are standing in front of ocean
BLEU value: 2.4

Image 4: Two people are walking on the road
BLEU value: 1.8

Image 5: brown dog and man on the air in ice
BLEU value: 2.4

We conclude that,

1. The performance of the model can be improved by training it on a larger dataset and hyperparameter tuning. 2. From above we can observe that unigram BLEU scores favour short predictions. 3. Prediction is good if all the BLEU scores are high.

BLEU = 1.94

Done By:-

- M. A. Hadi - IMT2018041
- G.V.Raghava - IMT2018023
- Sairama Shashank Kadiyala - IMT2018064
- Naveen Kumar - IMT2017029