Image Caption Generator

Shree Murthy, Rahul Sura, Dylan Inafuku

2023-12-16

Table of Contents

1. Introduction

This report aims to discuss the model building and evaluation of an Image Caption Generator trained on the Flicker 8k Dataset. The goal of this model is to take in an image and generate a caption that describes the image. This model is built using an Encoder/Decoder structure. The motivation for this model is to develop a model that can help create "Alt Text" to help visually impaired people understand images on the internet. This model can also be used to help people who are learning a new language understand the meaning of an image.

1.1. Alt Text

Alt text is a short description of an image that is used by screen readers to describe images to visually impaired people. The goal is to make the internet more accessible and useable for everyone. HubSpot has a great article on how to write alt text for images. It explains that alt text should be short yet descriptive. The goal of alt text would be to help people access wide variety of websites without worrying about not being able to understand the images. Over 2.2 billion people have a vision impairment or blindness. This is a huge number of people who are not able to access the internet in the same way as people who are not visually impaired. Furthermore, users who use a screen reader report they are ~61% less likely to retain all the information presented by an image. With this information in mind, we sought to create a model that could be helpful generating captions for images to help people who are visually impaired understand images on the internet.

2. Analysis

This section aims to discuss the dataset, data cleaning, and preprocessing. The dataset, again, will be the Flickr8k dataset. This section will also conduct some Exploratory Data Analysis (EDA) to help understand the pictures in the dataset as well as the format of the captions.

Note: The original creators provided a public domain license to enable anyone to reference the Flickr8k Dataset

2.1 Flicker8k Dataset

The dataset contains ~8000 images of various different stock images. These images have ~5 captions attached with them (located in the captions.txt file). First let's start by examining the images in the dataset.

2.1.1. Dataset's Images

Below are the randomly sampled images from the dataset:

Figure 1: Flickr8k Images

Based on the images above we can notice a few things. We notice a lot of pictures were taken outdoors (either in nature or in the city). This indicates that the model is mostly going to learn about the outdoor environment and related features, such as grass, buildings, the sky, and more. Furthermore, we notice a decent number of the images had animals. While this may not be accurate across the entire dataset, it is still vital information because it informs us that the dataset has variety when it comes to what's being depicted. The model shouldn't be biased to just humans, it requires a diverse set of information to help train it effectively and ensure it is knowledgeable about all of our world's features. Also, the images are augmented differently, they are all different shapes and sizes (some vertical and some horizontal). Moreover, the pictures have some blurry backgrounds and are taken during different seasons of the year, such as the one taken in the snowy terrain.

2.1.2. Dataset's Captions

Now, let's analyze the structure of the image captions. These captions are located in the captions.txt file and have at least 5 captions per image in the dataset. Below is a table with 5 random captions.

Image	**Caption**
1000268201_693b08cb0e.jpg	A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg	A girl going into a wooden building .
1003163366_44323f5815.jpg	a man sleeping on a bench outside with a white and black dog sitting next to him .

Image	Caption
1012212859_01547e3f17.jpg	"A dog shakes its head near the shore , a red ball next to it ."
1287475186_2dee85f1a5.jpg	Little boy sitting in water with a fountain coming up through his lap .
1287475186_2dee85f1a5.jpg	A boy sitting in water .

Based on the image captions above we can see there is a lot of variation happening, meaning there will be a decent amount of data cleaning taking place before we tokenize the captions. First we notice the captions aren't standardized, i.e they aren't all starting with capital letters, some have quotes and the others don't. This indicates that there wasn't much emphasis placed on creating standardized captions, rather they placed them into the document without any cleaning. Furthermore, the lengths of the captions are of varying degrees. Some are really short, such as the second caption and last caption. The differing lengths are extremely useful for our problem statement, we don't want captions to 'over stay their welcome.' Essentially, we want the captions to be short and detailed to ensure the features of the image are explained properly without any loss in information.

2.2 Data Cleaning

For data cleaning our outline was very clear. We didn't want a lot of noise with the actual words itself which may create a scrambled caption. We knew that our model wasn't going to create great captions, but we could at least make sure that the semantics of the caption itself (i.e. misplaced punctuation) could be kept at a minimum. The first step we took in isolating the captions was to create a dictionary with the keys being set to the image ids (i.e the filename(s)) and the values were set to the caption for that image id. The goal was to ensure that we can have a safe and secure way to just analyze the captions and not influence the model by passing in the image ids. The next step was to clean up the captions, essentially standardize, the words. This was achieved by our data cleaning function in the `model.py` file called `cleaning`. The method took a dictionary of image names and their corresponding captions, and then did miscellaneous things to it. Some of those things include removing punctuation, as mentioned, as well as making all the words lower case, trimming surrounding spaces. This will lessen the bias the model will have towards certain words and characters.

3. Methods

3.1 Preprocessing

One of the preprocessing steps, inspired from our CPSC-393 classwork, was adding two custom tokens/tags for each caption. Towards the beginning of the caption, we would add something called `'startseq '` and at the end, we would add `' endseq'`, still following the standardization conventions mentioned for the sake of consistency. These two tags enabled us to help train the model in order to recognize that at the beginning of the sequence, there is a logic-sounding start, and at the end of the sequence, there is a logical end. This mitigates the chances of the model generating a caption that starts or ends with something like the word "and". Since every single caption prepended and appended these two custom tokens uniformly, it reinforced the model to understand the start and end a little better.

4. Results

5. Reflection

When tackling the problem of image captioning, we first analyzed what we already knew. Throughout CPSC 393, we learned CNNs, LSTMs, and transfer learning models. However, we never really put all those ideas together (i.e Model Fusion) which was a necessary step for this project. The closest thing we learned to multiple models working together were transformers. With that initial knowledge, we sought to blend CNN and LSTMs together to create the image caption model. Also, we could use transfer learning to help with the mundane tasks like feature extractions. Once, we figured out this baseline we built an effective model that could generate captions for the images. Nevertheless, this approach still had some limitations. The dataset we used had over 8000 images; however, most of the pictures were of people or household animals. Furthermore, most of the people were white or had pale complexions. This could've easily poisoned our model training stage because it could assume that when it sees a person they are automatically white, or have a pale complexion. This is not the ideal when trying to create a model that can be used by everyone and for any type of pictures. Also, the pictures were mainly outdoors and people doing activities. This also limits the model's ability to learn because it may not learn what features are important to tell the interior of a house, school, workplace environment, etc. Our world has various activities and tasks that people and animals do. While, this model showcases the capabilities to train on an image dataset to generate somewhat accurate captions, it is not the best model to use for a wide variety of images.

To improve this model, it would be ideal to have two things: a training environment with access to a large number of GPUs. We trained this model on one NVIDA GPU, while it

trained quickly we didn't have extra GPUs to handle a model that could be trained on a larger image dataset (maybe like 100,000 images with various features).

We hope this model creates a simple baseline for future researchers to use a simple, yet efficient, model design and tweak the hyperparameters and dataset information to create a more robust and effective model that can be used by all to generate alt text at an effective rate.

Overall, given our problem statement, i.e generate alt text, we created a sufficient model that can hold up to the task. However, we recognize there are limitations to this approach and hope that future researchers, as well as ourselves, can improve upon this model.

6. References

References are placed in order of appearance in the report.

1. HubSpot. HubSpot
2. Sharif, et. al. Understanding Screen-Reader Users' Experiences with Online Data Visualizations