AUTOMATIC IMAGE CAPTIONING in DEEP LEARNING USING PYTORCH

Sayali Pisal
Computer Engineering Department
Jose State University
San Jose, USA
sayali.pisal@sjsu.edu

Sneha Patil
Computer Engineering Department San
San Jose State University
San Jose, USA
sneha.patil@sjsu.edu

Chunchen Lin
Computer Engineering Department
San Jose State University San
Jose, USA
chunchen.lin@sjsu.edu

***Abstract*— In this project, we solve one of the difficulty faced by blind people to be aware of their surroundings. The main idea of this project to generate a caption that explains what is the image is about using (Convolutional Neural Network) CNN, (Recurrent Neural Network) RNN and Pytorch. Thus, with accurate training and testing the model we could achieve a good accuracy that will be helpful for the blind people to overcome**

I. PROBLEM STATEMENT

Blind people do have a challenging life. The estimated blind or visually impaired people in the US is about 10 million. The assistance that can help them to sense their surrounding is in constantly demanding for helping them to live a better life.

There are various tools and navigation systems for visually impaired people, but they all have their limitations that provide far less information than human eyes do.

One such solution to existing problem is Image Caption generation.

II. INTRODUCTION

Recently, deep learning and natural language processing have sheen lights into this field. Researches have been focused on replicating the human ability of describing an image or videos in natural language, providing a rich set of details at a first glance.

For a human, it is an easy task to describe the contains of an image by just have a look at it, but it is very challenging for computer algorithm to do the same thing, even with currently very powerful deep learning method.

In a computer algorithm view of point, generating a image caption requires brief understanding of natural language processing and ability to identify and relate objects in an image.

The ability of verbal description of a scene has been tackled by bridging natural language processing, the science of understanding human language with computing vision, the science of automatic extraction and analysis of visual information.

Before deep learning become popular, image captions are done based on hard-coded features and well-defined syntax. The major shortcoming of this method is the limitation of the type of sentences it can be generated by any given model. The key contribution of machine learning to image caption is that the model built by machine learn is free of any hard-coded feature or sentence templates. It is a natural learning process, just like the human brain, although still far less mature than a human brain.

The main challenge for using machine learning for image captioning is the enormous variety of visual data, which makes it very hard to predict a-priori and only driven by data what could be interesting in an image and what should be described. Another challenge is that most of the data are noisy and cannot be directly used in image captioning mode

Before deep learning become popular, image captions are done based on hard-coded features and well-defined syntax. The major shortcoming of this method is the limitation of the type of sentences it can be generated by any given model. The key contribution of machine learning to image caption is that the model built by machine learn is free of any hard-coded feature or sentence templates. It is a natural learning process, just like the human brain, although still far less mature than a human brain.

The main challenge for using machine learning for image captioning is the enormous variety of visual data, which makes it very hard to predict a-priori and only driven by data what could be interesting in an image and what should be described. Another challenge is that most of the data are noisy and cannot be directly used in image captioning mode.

Video captioning is one of the most challenging tasks in machine learning, which needs to generate a natural language captions as well as to learn the visual information within the sequence of visual contents.

In many related work, video captioning are done with an encoder-decoder framework. An encoder is generally a convolutional neural network(CNN), which extracts features of the video. A decoder is usually a recurrent neural network(RNN), which generate caption of the video.

In our project, we first trained and tested a deep learning model for image captioning.

III. Methodology

The two main modules involved in implementation are:

A. Identifying the Image (Convolutional Neural Networks – CNN)

The basic idea is to train the artificial neural network with the images so that it can predict to which class the image belongs to. Firstly, CNN model involves a series of steps. Starting with convolution, pooling, flattening and last step is Full connection. It initially starts by extracting the features of the image such that the vector formed from such classification is transformed in such a way that it acts like a input to RNN/LSTM network. Classification of the image is very typical and most used in various applications. Self-driving cars requires this feature as well to identify objects moving at high rate of speed. If you want to submit your file with one column electronically, please do the following:

As already mentioned understanding the semantics of any given image though captured statically or by a moving object is the crucial part of the problem.

B. Language Involved in the Classification (Recurrent Neural Networks - RNN)

The image and be further resized and normalized. For language training we are using RNN (recurrent neural network). LSTM (long short-term memory) is an essential part of RNN. This step involves in identifying the various labels associated with the image. The labels can be actions, verbs, nouns and even vowels. All these labels derived out of the image are then stored and the target sentence for the image is formed.

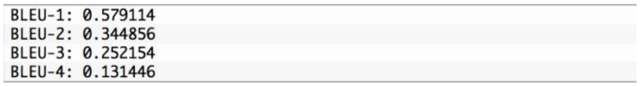
IV. IMPLEMENTATION

Numpy in python provides an array of dimension n and functions can be performed on this n dimension. It's a generic framework for calculations. It fails to be used in image captioning as it involves computing layers, graphs and gradient features. We decided to move tensor. Pytorch Tensor is identical to Numpy which can be used for performing any kind of computation. The important feature is that they utilize GPU to enhance the numerical operations. As deep neural networks are difficult to train. We faced a lot of challenges in doing so. Various datasets were used to get accurate results. We tried working on the Flickr dataset and finally settled on the MS-COCO dataset. This was solely since this dataset provided greater performance in identify the image. A significant increase in the object detection was seen. The implementation flow is given as –

1. Data Ingestion
2. Data Preprocessing
3. Deep Learning Model
4. Training the model
5. Evaluating Model
6. Generating New Captions

A. Data Ingestion

The dataset consists of images which are divided into training dataset containing 6,000 images, development dataset containing 1,000 images and test dataset with 1,000 images. BLEU scores are used to evaluate the skill of the model dataset, which is given as:



```
BLEU-1: 0.579114
BLEU-2: 0.344856
BLEU-3: 0.252154
BLEU-4: 0.131446
```

B. Data preprocessing

Feature extraction gives the internal representation of the image right before the classification is made. A dictionary of image features is returned resulting in a features.pkl file.

Each photo is assigned with a unique identifier. Each photo maps to one or more textual descriptions from the vocabulary file.

Cleaning the vocabulary file is done by following methods:

1. Converting words to lowercase
2. Removal of stop words
3. Removal of character words and punctuation
4. Removal of numeric data

C. Deep Learning Model

This section is divided into following three parts:

1. Data Loading

The image and text datasets are fed to the model for training where it also monitors the performance of the system and saves the generated caption for each photo file. The series of previously formed words will be provided as next input. Encoding the textual description of an image to numbers is performed to compare it to the model's future prediction which consistent mapping from words to unique integer values. Sentence splitting into words is done. Inputting the model with first two words, the next word will be generated by the model itself which will be encoded as integers and fed to the word embedding layer. Thus, the model outputs a prediction as a probabilistic distribution over all words in the vocabulary.

2. Model definition

Processing of the model is done by:

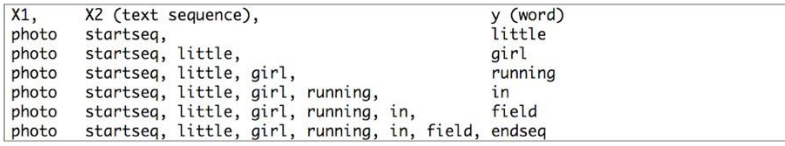
A. Extraction of features from photo
B. Processing Sequentially
C. Feeding the result to a decoder

3. Model fitting

Model fitting is done over an iteration of 20 epocs.

D. Training the model

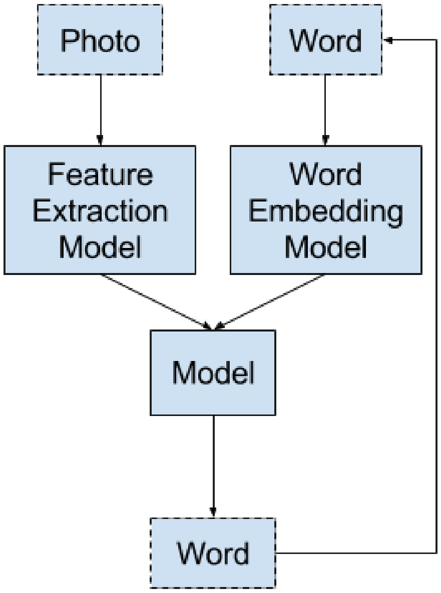
Training the models randomizes the order of photos with every epoch. Each photo is assigned with a unique id with a textual reference and saved back to the memory. Yield of more than one or more photos worth of all the samples is stored per batch. Thus, it improves the model performance with each processing epoch.



```
X1,     X2 (text sequence),                          y (word)
photo   startseq,                                    little
photo   startseq, little,                            girl
photo   startseq, little, girl,                      running
photo   startseq, little, girl, running,             in
photo   startseq, little, girl, running, in,         field
photo   startseq, little, girl, running, in, field,  endseq
```

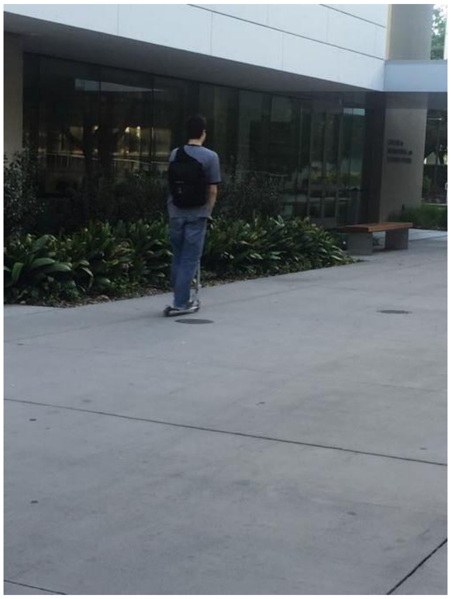
E. Evaluating model

Evaluation of model is done by describing the description for all the photos in the test dataset and evaluating those with a standard cost function. It involved parsing the character token strings calling the model recursively till the end of file is reached. The BLUE scores are used for comparing the translated text against reference translations.



F. Generating New Captions

Caption generation for real time photos is done. A tokenizer is used for encoding the generated words for the model while generating a sequence. Text to speech conversion needs to be implemented using API by IBM speech-synthesis capabilities.



Man standing on skateboard in the middle of a street

V.TOOLS AND ENVIRONMENTS

Python environment with :

1] Keras (2.1.5 or higher)
2] Scikit-learn
3] Pandas
4] NumPy
5] Matplotlib
6] Pytorch

VI. Future Implementation

Encouraged by our initial results, we are planning to deploy our image and video captioning application into a smartphone for the convenience for blind people. They can use the camera of the smartphone to capture images and videos of the surroundings. Text to speech-synthesis to be implemented by using IBM API

The machine learning model can caption the images or video in a very fast fashion. We also working on to build a real-time video captioning system. Since an app on a phone, the blind people must hold the phone to capture the video, we are planning to integrate the camera and the software to a smart glass, which capture and process the images in real time.

VII. Conclusion

We could analyze the models and experiment with different datasets to train the model to get a high accuracy that correctly identifies the objects in the environment.

VIII References

[1] https://ieeexplore.ieee.org/document/7840928/
[2] https://arxiv.org/abs/1512.03980
[3] https://arxiv.org/abs/1512.03980
[4] https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/
[5] https://towardsdatascience.com/visual-attention-model-in-deep-learning-708813c2912c
[6] https://www.semanticscholar.org/paper/Automated-Neural-Image-Caption-Generator-for-People-Elamri-Planque/cc61cd90529fede6e1bfb14042d021bc2a076e99
[7] http://www.cs.ubc.ca/~minchenl/doc/ImgCapGen.pdf
[8] https://en.wikipedia.org/wiki/Recurrent_neural_network