Automatic Image Captioning for visually impaired

Domain Background

Computers now are getting better and better with the help of modern discoveries and technological advances in field of Computer Vision with the development of modern Model Architectures and algorithms and the Rise of GPU Processors we can now use Computer Aid to help us in a lot of different activities

As Computers can now interpret Images and extract Features from it using Convolution Neural Networks automatic image Captioning is one the application of CNN's that can help us in describing images and scenes that can give better context for visually impaired people or help search engines like Google to better provide more relevant results to users search queries of images or advance the filed of self-driving cars by letting the car understand the scene in-front of it

In this proposal I go through achieving one of the goals of automatic image captioning by providing a software that can help people

History[1]

Over the years there has been a lot of development in the methods used for image captioning and here are some of the techniques that was previously used

- Retrieval based image captioning:
  - Given a query image, retrieval based methods produce a caption for it through retrieving one or a set of sentences from a pre-specified sentence pool. The generated caption can either be a sentence that has already existed or a sentence composed from the retrieved ones.

- Template based image captioning
  - another type of methods that are commonly used is template based. In template based methods, image captions are generated through a syntactically and semantically

[1] https://press.liacs.nl/students.mir/inspiration/A%20survey%20on%20automatic%20image%20caption%20generation.Neurocomputing2018.pdf

constrained process. Typically, in order to use a template based method to generate a description for an image, a specified set of visual concepts need to be detected first. Then, the detected visual concepts are connected through sentence templates or specific language grammar rules or optimization algorithms to compose a sentence

- Deep neural network based image captioning
  - Due to great progress made in the field of deep learning recent work begins to rely on deep neural networks for automatic image captioning
  - Encouraged by advances in the field of deep neural networks,instead of utilizing hand-engineered features and shallow models like in early work, deep neural networks are employed to perform image captioning. With inspiration from retrieval based methods

- Image captioning based on the encoder-decoder framework
  - Inspired by recent advances in neural machine translation the encoder–decoder framework is adopted to generate captions for images
  - an encoder-decoder architecture will be used in the project

Problem Statement

With the help of AWS sagemaker and Deep CNN architecture like VGG we will build a web-based software that can take an image from a user and provide caption for it in Arabic Language and using Google API, speaking the predicted caption out loud.

Problem Solution

The solution to the problem can be divided to two parts

1. **Computer-Vision:** given a data-set of of images,captions pairs we will use a pre-trained VGG model as an Encoder model in addition to some LSTM Layers as decoder network and fine tune the architecture on our data-set and optimizing it to minimize a cross-entropy Loss to predict a caption for our image employing techniques like word embedding basic NLP text operations

2. **Software:** using sagemaker the model can be deployed to an endpoint which can be queried and allows us to have a backend to serve our software

Datasets and Inputs

For our project I choose the flicker8k data-set for image captioning, Original data file of the flicker contains

- **Flicker8k**: contains 8091 jpg images each image has 3 corresponding captions in Arabic
- 6000 images are in train,1000 in dev,1000 in test set

Original flicker8 data-set with English captions can be found on Kaggle here

For the Purpose of our project thanks to this Paper we can have the same data-set with the same images but the captions are in Arabic

Data Sample

شخصان يجلسان على رصيف
خشبي على بحيرة عند غروب الشمس

رجل وكلب على مقعد في الحديقة

[2]

Benchmark Model

We will take the encoder-decoder model built in this paper which is the paper we obtained the data from as benchmark for our work in project as this model is trained on the same data we will train our model on.

---

[2] **data citations**: M. Hodosh, P. Young and J. Hockenmaier (2013) "Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics", Journal of Artifical Intellegence Research, Volume 47, pages 853-899

The authors model managed to get a Bleu-Score of 0.35 so this what i will base the model performance on.

Evaluation Metrics

BLEU (BiLingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text (in our case Predicted Captions) to a set of high quality reference translations(in our case label captions in our data-set) predictions on the test Set can be evaluated by using Bleu-Score
Bleu-Score formula:

$$
\begin{aligned}
Geometric\ Average\ Precision\ (N) &= exp(\sum_{n=1}^{N} w_n\ log\ p_n) \\
&= \prod_{n=1}^{N} {p_n}^{w_n} \\
&= (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}}
\end{aligned}
$$

$$
Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}
$$

$$
Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)
$$

Project Design

Project design consists of 4 major steps:

1. Preparing Data
    a) Reading data images and captions from text files
    b) Prepossessing the text captions by removing punctuation's and arabic Stop words.
    c) Transforming words to indices and vectors fit for the model

2. Modeling

a) Loading a pre-trained VGG16 model without top Softmax Output Layers
b) Adding LSTM Layers and Dropout Layers to predict captions
c) Writing the code to train.py
d) Using a sagemaker Tensorflow Estimator to use train.py as entry-script
e) Fitting the model on train and validation set to optimizer cross-entropy Loss

3. Testing
    a) Testing the model on the test set and evaluating using bleu score
    b) Deploying the model to a sagemaker endpoint
4. Pipeline
    a) A preprocessing lambda for preprocessing.py script
    b) A train lambda that loads the trained model on sagemaker to train new data
    c) An inference lambda that invokes the endpoint to get predictions
5. User-interface
    a) Building a user-interface to send images to the API and get predictions