Caption this Pic

Statement of Work

Harvard AC215
Group name: pinkdrink
Anita Mahinpei, Jianhui Tang, Benjamin Liu, Yingchen Liu

Problem Definition & Background

The World Health Organization (WHO) estimated that 314 million people have visual impairment across the world, including 269 million who have low vision, and 45 million who are blind (Ono et al 2010). Many people with visual impairments rely on screen readers in order to access the internet through audio and thus depend on image captions (Yesilada et al 2004). Therefore, accessibility, as well as automatic indexing and other goals, make accurate image captioning an important priority (Hossain et al 2018).

Proposed Solution & Scope

We will explore the realm of image captioning by creating an app that allows users to upload images and have them be captioned in parallel through multiple models. We specifically focus on creating captions for images of objects and scenery. The app will generate and display several possible captions for the image. Since the intended audiences are visually impaired individuals, the app will provide a text to audio functionality so that the generated captions can be read out loud to the user if desired.

Our models will be trained with a dataset of more than 330k images with 1.5 million object instances, covering more than 80 entity categories. We will explore a full-stack data science process starting from deep learning, operations to deployment and eventually distribute the project app in a contained environment to a wider range of audiences.

Data & Model

We will use the Microsoft Common Objects in Context (COCO) data for our project. COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)

- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

In terms of deep learning models, both computer vision and natural language processing models will be used. As an initial step, we will set up a simple CNN encoder - RNN decoder baseline model. Then we will explore using more complex, pre-trained architectures for the encoder and decoder part of the model. We will use and fine tune to our COCO dataset:

- **Computer Vision:** Pretrained CNN architectures such as DeepNet, VGGNet, ResNet, and DenseNet for feature extraction.
- **Language:** Pretrained language embeddings and transformer-based models such as BERT/GPT for text generation.

We will also explore at least one of the SOTA models in the current image captioning leaderboard (https://paperswithcode.com/sota/image-captioning-on-coco-captions). Since images in the wild do not typically have accompanied annotations/auxiliary inputs, we are particularly interested in SOTA architectures that do not use auxiliary inputs like the Meshed-Memory Transformer Model by Cornia et al. This is a fully-attentive model inspired by transformer models for machine translation. It consists of a multi-layered memory-augmented encoder followed by a multi-layer decoder with a meshed cross-attention mechanism which allows for incorporating both high level and low level image features.

Since the models will output natural language sentences, we cannot use standard model evaluation mechanisms such as F1-score or accuracy. We will be using metrics that are commonly used for evaluating machine translations such as BLEU, METEOR, ROUGE-L, and CIDEr.

Potential Timeline and Components

Sprint ending	**Tentative milestone or goal**
9/23	**High Level Picture** ● Milestone 1 - Statement of Work ● Brainstorm possible solutions ● Identify model tasks **Data Pipeline** - Store the COCO dataset in a GCS Bucket so various team members can access and work on the same data - Start with a subset of the data (~10k images) and zip it similar to the original dataset - Spend some time to understand the data and do some EDA

10/7	**Modeling** - Research SOTA models for the image captioning task such as those mentioned in recent image captioning survey papers (e.g. https://par.nsf.gov/servlets/purl/10113614**)** - Use the data subset to build our data downloading and data processing pipelines (e.g. resize all images to the same size, remove noise, normalize the inputs, create a data augmentation pipeline, etc.) - Do a quick proof of concept (POC) of our baseline (CNN-RNN) model and various SOTA models that could be used. Use the subset data to prove out the concept and make sure we can get everything to train. - Finally train on the original dataset - Save training weights and parameters after each run/experiment of the model to a GCS Bucket
10/22	**UI / App Design** - Start coming up with and wireframing a high-level design of the App - Identify containers, frameworks, components required for the AI App - Create the GitHub repo structure for our codebase for the AI app (app frontend, app backend, services, deployment scripts, etc.) - Look for open-source text-to-voice APIs that could be used in the app for reading the captions out loud - Start with a simple flow of components to get everything working together (data, model, app frontend, app backend)
11/15	**Deployment** - Deploy containers to GCP to make sure everything works in “production” mode - Create scripts to scale our app using Kubernetes and deploy to GCP
12/5	**Final Deliverables and Presentation** - Wrap up the code and documentation - Slides for presentation - Final deliverables - Scalable deployment of AI App using Kubernetes