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, have them captioned in parallel through multiple models, and read out the captions to users. 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.

The scope of our model will be built upon training data over more than 330k images with 1.5 million object instances, covering more than 80 entity categories. We will explore a full-stack data science practical 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 CNN encoder - RNN decoder baseline model.

- **Computer Vision:** Pretrained Image models as feature extractors, such as DeepNet, VGGNet, ResNet, DenseNet, fine tune to fit our COCO dataset.
- **Language:** Pretrained language embeddings or transformer-based models such as BERT/GPT for text, fine tune to fit our COCO dataset with captions.

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 dataset in a GCS Bucket so various team members can access and work on the same data - Start with a subset of the data and zip it similar to the original dataset - Spend some time to understand the structure of the database
10/7	**Modeling** - Research SOTA models / other model implementations for the model tasks - Use the subset to build your data downloading and data processing pipelines - Do a quick proof of concept (POC) of the various models that could be used. Use the subset data to prove out the concept and make sure you can get everything to work, train, etc - Finally switch to the original dataset

	- Save your training metrics after each run/experiment of the model. It will be easier to save your metrics across team members run in a single data store like a GCS Bucket
10/22	**UI / App Design** - Start coming up with a high-level design of your AI App - Identify containers, frameworks, components required for your AI App - Create a GitHub repo and come up with a folder structure for your codebase for the AI app (app frontend, app backend, services, deployment scripts, etc.) - 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 your app using Kubernetes and deploy to GCP
12/5	**Final Deliverables and Presentation** - Wrap up everything - Slides for presentation - Final deliverables - Scalable deployment of AI App using Kubernetes - Everything to Github