Milestone 2 Project Report

Current State of the project

App Development

Two more activities have been added to the app. The functionality for making a meme from a given caption is complete; however, functionality for making a meme from a given image needs debugging (to be completed for the next milestone). Currently, we have added 40 images (10 per cat emotion) to a folder inside the app project so an image can be randomly selected based on the sentiment analysis of the caption provided by the user. Furthermore, our app is now able to access images from the Android gallery whereas the permission was restricted before. This means that if the user's gallery is populated with images, the app should be able to pick an image from the user's library to combine it with a randomly selected caption and create a meme. The caption to be selected at random is part of a list of captions (10 per cat emotion), all of which are somewhat related to CPEN and our 2nd year experiences, as created by us.

Source used for Image Uploading button:
https://www.geeksforgeeks.org/how-to-select-an-image-from-gallery-in-android/

Text Classification

We modified some categories in the original dataset to fit our app, the original dataset consisted of the emotions: fear, surprise, love, joy, sadness and anger. We removed the entries with the sentiment surprise, changed the categories fear to sad, love to sleepy, joy to happy, and sadness to sad. The original dataset was in separate files for testing, training and validation in the text format, we changed that to a single CSV file. All the dataset manipulation was handled using Pandas framework.

For the Sentiment analysis: We created a dataset class that would handle the CSV data, by creating tokens and the dictionary, and dividing the data into training and testing datasets. Then we use another class to create offsets in our dataset in order to create a tuple of tensors of labels, caption text, and offsets. Then we used an embedding bag to create embeddings. Finally, we used SDG as our optimizer with a learning rate of 3.0 and within 10 epochs we were able to get training loss: 0.005710, training acc 0.9809, test loss 0.015266, test acc 0.9149. We also tried to use the ngram to analyze the sequential information, but the accuracy was about 0.8, so we kept our first implementation.

Image Classification

We fixed some errors in the image classification part, and optimized the parameters for scheduler and optimizer. The accuracy is around 70% now. We are planning to try removing the animated pictures or cropping out the captions on the images to see if it will bring the accuracy to 80%.

Current Challenges and Bottlenecks

The integration of our PyTorch models with the app remains a challenge. We plan on using PyTorch Mobile to integrate our ML models into the app. Also we need to test our Image Upload buttons and check if they give desirable results.

For Milestone 3, we plan to work on the integration of ML models with the app and general fine tuning of the app. Improving aesthetics would be a large part of what we do in the last two weeks.

Work Distribution

Christy: Worked on collecting 20 images for the app's dataset, generated 10 captions for the "happy" emotion. Collaborated with Jayant on the text classification task. Fixed errors and made optimizations on image classification.

Fariha: Worked on collecting 20 images for the app's dataset, collaborated with Savitoj on the app development, and generated 10 captions for the 'Sleepy' emotion

Jayant: Worked on collecting 20 images for the app's dataset, generated 10 captions for the "sad" emotion. Collaborated with Christy on the text classification task.

Savitoj: Worked on collecting 20 images for the app's dataset, collaborated with Fariha on the app development, and generated 10 captions for the 'ANGRY' emotion.