Dillon McCarthy
DS 552

Final Project Description

Project Title:
CaptionGen

Introduction and Objective:
This project will allow users to generate social media captions for a specified input image. The model will analyze the image and then generate creative text to accompany the image on a post, streamlining the social media process. This would offer small businesses and personal brands a faster process to generate social media posts, provide scalability for frequent posts, and consistency on brand. This generative application is very applicable for the social media domain and fills a current gap in the field of content generation for businesses and professional use cases.

Selection of Generative AI Model:
Generative modeling will be used for transformations from image-to-text and text-to-text in a multimodal application. The models used were:

- BLIP
- GPT family (GPT-2 and GPT-4o)

BLIP is chosen for being one of the leading models in image captioning. Social media images will be diverse and require clear labeling, so this pre-trained model can be fine-tuned on images specific to social media to get an accurate analysis of the input photo. Natively, this caption is not creative and usually provides concise and straightforward captions, so it will then be handed off to a GPT to generate the post as a text-to-text input. The GPT family will be used due to its industry leading performance and creativity. GPT-4o was chosen for its creativity and human-like writing capabilities spanning a diverse vocab including emojis that are imperative for modern social media. Additionally, GPT-2 was chosen for its availability as a pre-trained, open-sourced model as a candidate for fine-tuning. The combination of these models and their relevance in current research make them a strong choice for the application

Project Definition and Use Case:
These models will be utilized for content generation in a social media image captioning system. This domain offers creativity and increased efficiency for all, but in particular small businesses and personal brands will benefit the most. The BLIP vision-language model will be paired with the GPT large-language model to offer a simplified solution for generating social media posts. There are many use cases for social media, but today it serves as one of the best marketing channels for many businesses. Different platforms require different types of content, and maintaining consistent and frequent content can be challenging. Not only that, but being creative with social media posts is what onboards consumers onto a product or service.

This project will offer custom social media content generation tailored to multiple platforms and styles. By supplying an image with a platform type and predefined style, the application will output an appropriate caption.

Platforms	Styles
● Instagram ● X (Twitter) ● Facebook	● Engaging ● Professional ● Funny

Table 1: CaptionGen input parameters

Additionally, other model parameters will be visible and customizable to the user including temperature, top_p, and max_output_tokens which are directly fed into the models. The combination of all these parameters produce several unique themes in which users can experiment with and maintain their preference.

Furthermore, for longer inference time users can optionally enable one-shot captioning using CLIP. While the name may suggest faster inference time, this feature will generate four captions using four different sampling techniques: greedy, beam search, top-k, and top-p. Then, these will be fed into CLIP and converted to text embeddings and compared against the image embedding over the cosine similarities to select the highest ranking caption. This optional feature is disabled by default, but can enhance accuracy with more challenging images.

Implementation Plan:
This application was developed using the following technology stack:

Libraries and Frameworks:
- *Hugging Face Transformers*: model architecture, tokenization, and training
- *Hugging Face Datasets:* load and process datasets efficiently
- *PyTorch*: deep-learning framework for training the model
- *PIL:* pre-process and load images
- *GPT-4o:* used to generate a synthetic Instagram caption dataset

Compute Resources:
- *Google Colab:* short-term prototyping and experimenting with different models and training techniques
- *WPI Turing Cluster*: longer-term training jobs greater than 30 minutes using an NVIDIA A100 GPU

Models:
- *BLIP (Salesforce):* pre-trained image captioning model, fine-tuned on datasets
- *CLIP (OpenAI)*: pre-trained zero-shot transfer learning model
- *GPT-4o (OpenAI)*: pre-trained multipurpose model accessed through API

- *GPT-2 (OpenAI)*: pre-trained model open-sourced and fine-tuned on datasets

Data Storage and Management:
- *Hugging Face Datasets:* central location for accessing large datasets
- *Google Drive:* private storage for private dataset
- *Parquet*: streaming subsets of data for manageability

Web Application and User Interface:
- *Streamlit*: create and deploy a front-end web interface to interact with the models
- *GitHub*: share code and details regarding the design of the application

The following design process was used to develop, test, and deploy the application:

Develop:
1. Design model architecture
2. Identify pretrained models
3. Identify several datasets
4. Fine-tune BLIP and GPT-2
5. Develop back-end framework to load models
6. Integrate back-end with CLIP
7. Integrate back-end with OpenAI APIs for GPT-4o
8. Add style prompts to GPTs
9. Input/output text cleaning and processing

Test:
1. Evaluate individual model performance using human-evaluation
2. Adjust hyperparameters and repeat: epochs, learning rate, batch size, temperature, top-p, top-k, max length, warmup ratio, repetition penalty.
3. Using best results, train long term on Turing
4. Peer feedback

Deploy:
1. Remove API key from code
2. Create BLIP and GPT models with their access token in Hugging Face
3. Upload BLIP and GPT tensor weights to Hugging Face using access token
4. Adjust code to use Hugging Face model instead of loading locally
5. Test loading model from Hugging Face with Streamlit
6. Upload remaining code to GitHub (Can't store large files like the models)
7. Verify requirements.txt for Streamlit to install dependencies
8. Deploy new Streamlit App with GitHub repo and OpenAI secret
9. Monitor Streamlit Community Cloud App install all dependencies and build the app
10. Share the URL and let others start captioning their social media posts!

Model Evaluation and Performance Metrics:

The application architecture is non-deterministic and has two independent components that could be evaluated. This makes it challenging to evaluate the performance on specific metrics that are commonly used such as ROUGE and BLEU since these metrics require a reference dataset. Nonetheless, BLIP was evaluated using unseen validation data from Obscure-Entropy/ImageCaptioning_SmallParquets to compute ROUGE and BLEU scores.

Metric	CaptionGen BLIP
ROUGE-1	0.2275
ROUGE-2	0.1230
ROUGE-L	0.1989
ROUGE-Lsum	0.1988
BLEU	0.0017

While the model scores low for ROUGE and BLEU scores compared to industry standards, that does not necessarily reflect model performance. The key evaluation metric for this application is creativity, and that is not something that can be measured with a number. The generated captions will not have much N-gram overlap with the reference captions, and that is by design. Instead, the model is trained to output longer and more descriptive captions capturing the creative and nuances of the dataset captions.

Furthermore, when evaluating the application on an end-to-end basis, the model performs extremely well. The following are sample captions were generated on Streamlit:

Strolling through paradise with my furry sidekicks! 🐾✨ Who's leading who? #DogAdventures #NatureBuddies

🌿✨ Dive into the garden vibes! Lush greens, sunshine, and smiles all around. Who's ready for some outdoor fun? 🌼🎉 #GardenParty #GoodVibes

Deployment Strategy:
Streamlit offers its Community Cloud platform for hosting and sharing Streamlit applications. Once CaptionGen was fully functioning locally, the deployment process was initiated to share the app. Several steps were identified to deploy and publish the code:

1. Remove API key from code
2. Create BLIP and GPT models with their access token in Hugging Face
3. Upload BLIP and GPT tensor weights to Hugging Face using access token
4. Adjust code to use Hugging Face model instead of loading locally
5. Test loading model from Hugging Face with Streamlit
6. Upload remaining code to GitHub (Can't store large files like the models)
7. Verify requirements.txt for Streamlit to install dependencies
8. Deploy new Streamlit App with GitHub repo and OpenAI secret
9. Monitor Streamlit Community Cloud App install all dependencies and build the app
10. Share the URL and let others start captioning their social media posts!

Expected Outcomes and Challenges:

Oftentimes in generative modeling applications one of the greatest challenges is access to GPUs for training and compute workloads; however, the limiting factor for this application was the data. Originally it was expected that training and fine-tuning models for coherent and appropriate output would be challenging, but aggravated by the lack of high-quality, accessible, and manageable data for the use case. For this project there were three opportunities for training:

i. Image to Long-Text: enhance image captioning to directly transform image weights into social media-worthy captions
ii. Short-Text to Long-Text: enhance a short caption into a longer, more creative caption catered for social media
iii. Image to Short-Text: enhance image captioning to more descriptively identify objects within images

The first approach would limit the loss of information when transforming images into text, and by fine-tuning on real social media images and captions the better results the model could produce. Nonetheless, while many Twitter datasets exist, these are generally just text based. Since visual datasets were needed, two were identified: Instagram Influencer Dataset and "takara-ai/image_captions" from Hugging Face. The former was a private dataset with access granted from the researchers to a Google Drive containing ~200GB of image-caption pairs spanning 64 multipart zip files. After several attempts to work with the dataset, including both Linux, Windows, and Python decompression tools, the multipart zips were unable to be loaded for training. Similarly, for the Hugging Face dataset, the size proved challenging to work with and exceeded Google Colab's disk size for the free tier. Combined attempts to work with these datasets intermittently absorbed almost two weeks of development.

This left the second option: training a text-to-text model on enhancing a short caption into a longer, more creative caption for social media. After heavy research, the Waterfront/social-media-captions was identified as an ideal match for the task. This dataset contains 45,400 pairings of “human” and “assistant” captions, where the assistant expands upon the human caption to make it suitable for social media. Unfortunately, despite various hyperparameters and fine-tuning techniques, the models did not evaluate well after training. Captions generated were off topic and hallucinated objects or people, and included an excessive amount of hashtags. The challenge here lay with the quality of the data itself, and without any images in the data, it seemed a lot of details were lost between the human and assistant captions.

This left the final option of enhancing a basic image captioning model with traditional image captioning datasets. These are plentiful, but to capture the most details the Obscure-Entropy/ImageCaptioning_SmallParquets dataset was identified as a strong candidate to enhance BLIP’s captioning details. The dataset contains 1.5 million small image-caption pairs at around 34GB, and many rows contain several adjectives and descriptive identifiers of objects in the image, something the pretrained BLIP model does not always capture.

Resources Required:

The following lists the required resources for developing the project.

- Python, PyTorch, Hugging Face Transformers, Hugging Face Datasets
- Google Colab T4 GPU runtimes (free but limited)
- WPI Turing Cluster
- Streamlit
- OpenAI API Key (optional but recommended for best performance)

Conclusion:

The application performs very well and quickly generates captions. While it is challenging to evaluate without a defined target dataset, human evaluation shows that using the GPT-4o model with BLIP gives exceptional performance. GPT-2 does not perform that well and does not always generate coherent or meaningful captions. Despite the architecture of two pre-trained models, the inference time is strong and usable when running locally with around 5 seconds to generate a caption after the image is loaded. Unfortunately, for a free web hosting service like Streamlit, one of the drawbacks is significant overhead for resources which aggravates inference times. When generating a caption with Streamlit Community Cloud, end-to-end inference time takes closer to 30 seconds.

Altogether CaptionGen offers a streamlined method to rapidly generate creative captions for images with consistent styles. GPT-4o integrates extremely well with the fine-tuned BLIP model and is highly recommended for use. The application creatively engages readers through interesting captions, enhancing the visual storytelling experience and making content more compelling and relatable. Whether for social media, marketing, or personal projects, CaptionGen ensures that each image is accompanied by a captivating and contextually relevant caption, elevating the overall impact of the visual content. Go see for yourself!