Domain and background

I will be focusing on text generation for this capstone project It is an interesting method which leverages various deep learning algorithms to output a text given input as a text. It involves training these models on large datasets of text to learn patterns, grammar, and contextual information. These models then use this learned knowledge to generate new text based on given prompts or conditions.It is an extremely interesting and useful field of study that has many practical applications including but not limited to

- Content generation
- Conversational AI
- Paraphrasing
- Summarization

Problem statement

One interesting tangential field that endlessly fascinates me is text-to-image models like stable diffusion, DALLE etc. A crucial aspect of the operation of those models is prompting. The right set of prompting often makes or breaks the end result. Also often times the general and non-technical people often cannot express their needs succinctly for those text-to-image generation models to work as expected leaving them with dissatisfied results.

In this project, I aim to reduce that pain point by building a text generation model that can generate high quality prompts, ready to be ingested by text-to-image models. More formally I aim to build a text generation model that given an input word or phrase, generates a detailed and relevant prompt to be used by text to image models like Stable Diffusion.

Example: if I input *woman portrait* it should generate a relevant prompt like: *Portrait of a beautiful woman with long hair on bicycle with a Vangog style.*

Datasets and inputs

Text generation models rely heavily on the input data used to train the model. Hence a quality dataset is paramount to the success of this task. Aside from being high quality, the dataset should also be sufficiently large to capture the entire diversity of related text of this domain.

I selected this https://www.kaggle.com/datasets/tanreinama/900k-diffusion-prompts-dataset publicly available dataset to train or finetune my system. The dataset meets all the above mentioned criteria and has been used extensively by the community for related tangential tasks such as prompt classification, retrieving prompt from image etc.

Solution statement

I would develop a text generation model that can generate relevant prompts. These models can be both probabilistic models such as bag-of-words, markov chains and also deep learning based models such as RNNs, LSTMs and more sophisticated language models like transformers and llms.

Keeping up with recent developments in this problem domain I will opt towards deep learning based solutions as they offer excellent accuracy and relevant results. I would train these deep learning models with only the prompt of the mentioned dataset. The prompts would additionally need to be extracted, preprocessed and made into a format to be ingested by such models.

Once the data is ready, I aim to approach the task in a two fold way. I want to first train a vanilla LSTM on this dataset and for a more robust solution I would finetune a transformer based model like DeBerta or GPT to generate the prompts. For the sake of simplicity and resource constraints I am choosing to skip training a transformer from scratch.

Benchmark

The vanilla LSTM that would be pretrained on this task would serve as the baseline or benchmark for this task. Since it is the simplest and an early precedent of transformer based models and also widely used for text and character generation it's results can be reliably used as a starting point for improvement.

Evaluation Metric

Cross Entropy - Cross Entropy is a metric that calculates the difference between two probability distributions. Each probability distribution is the distribution of predicted words

Perplexity - The Perplexity metric is the exponential of the cross-entropy loss. It evaluates the probabilities assigned to the next word by the model. Lower perplexity indicates better performance

Project design

Keeping up with recent developments in this problem domain I will opt towards deep learning based solutions as they offer excellent accuracy and relevant results. I would train these deep learning models with only the prompt of the mentioned dataset. The prompts would additionally need to be extracted, preprocessed and made into a format to be ingested by such models.

Once the data is ready, I aim to approach the task in a two fold way. I want to first train a vanilla LSTM on this dataset and for a more robust solution I would finetune a transformer based model like DeBerta or GPT to generate the prompts. For the sake of simplicity and resource constraints I am choosing to skip training a transformer from scratch.

Pipeline specifics

The data would be downloaded and stored on S3. Since the text preprocessing would be different for the kinds of architectures I mentioned, I aim to make the process configurable and extensible to cater to various deep learning models by leveraging Sagemaker’s batch transform and related data processing pipelines. I aim to use Sagemaker as the primary source ecosystem to train, evaluate and deploy the solution as a managed endpoint.

The endpoint will be invoked via lambda function from a simple fastAPI webserver hosted on a t3.micro EC2 instance that users can interact with.Data Flow:

User Flow:

User: Interacts with the FastAPI webserver.

FastAPI Webserver:

Receives user input.

Triggers a Lambda function.

Lambda Function:

Invokes the SageMaker endpoint with the user input.

SageMaker Endpoint:

Processes the input using the trained ML model.

Returns the model prediction to the Lambda function.

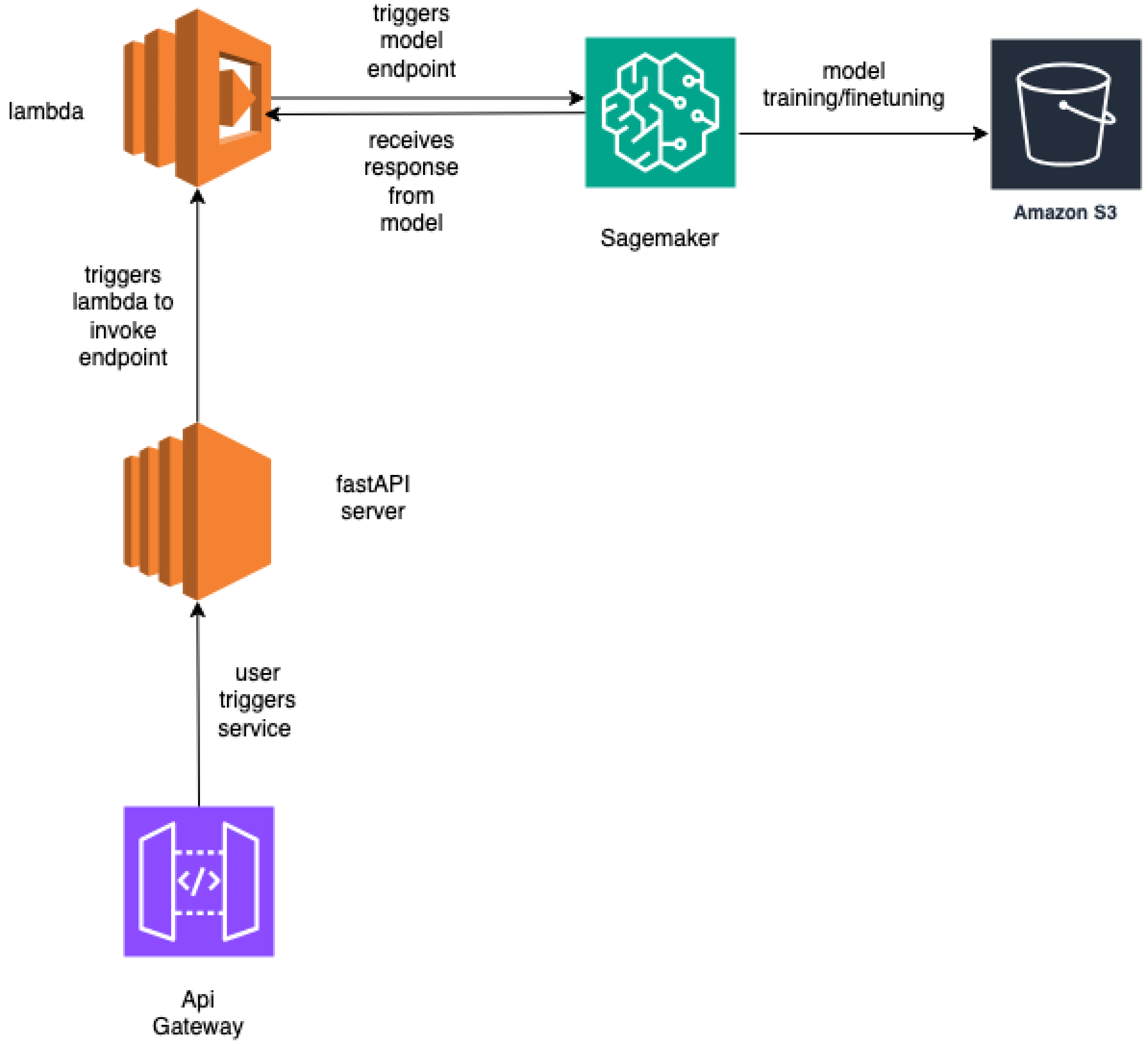
Lambda Function:

Sends the model prediction back to the FastAPI webserver.

FastAPI Webserver:

Displays the model prediction to the user.

Architecture Diagram


lambda
triggers model endpoint
receives response from model
Sagemaker
model training/finetuning
Amazon S3
triggers lambda to invoke endpoint
fastAPI server
user triggers service
Api Gateway