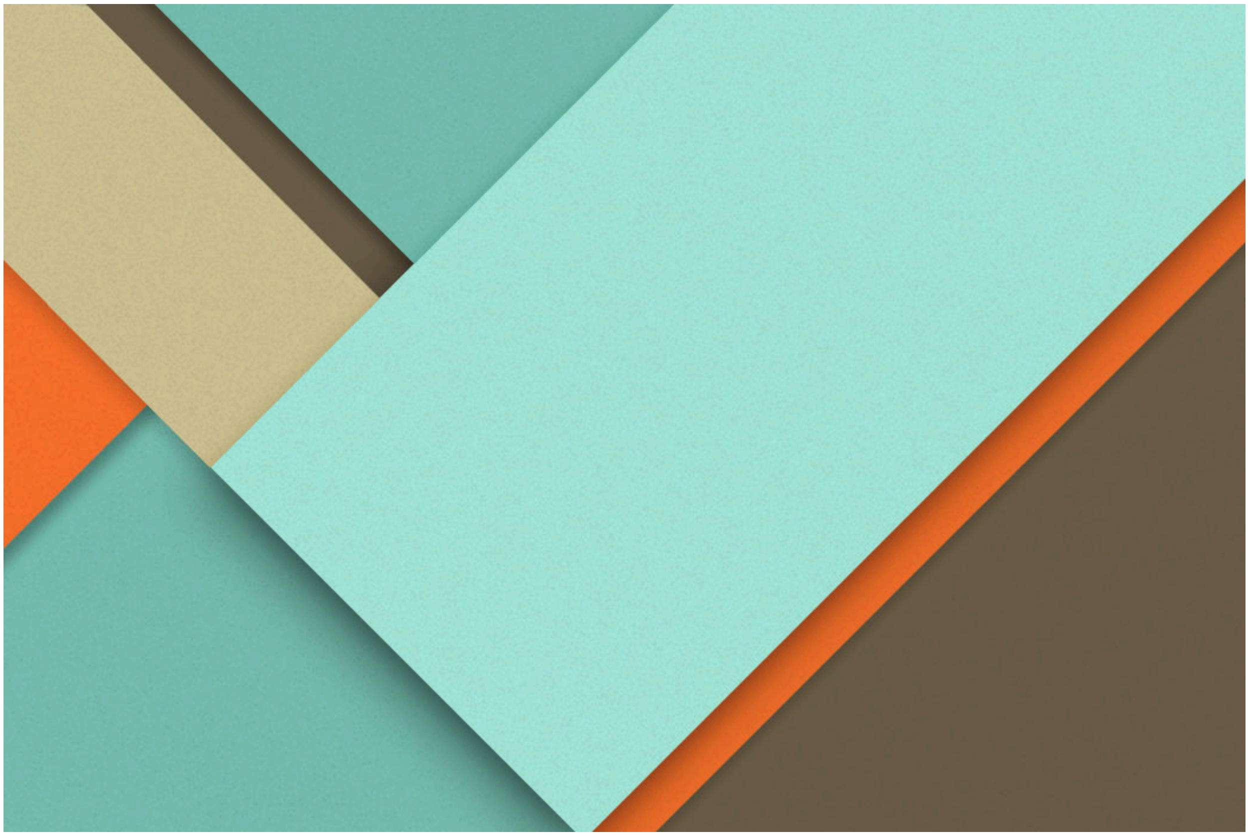

AI-based Generative QA System

29.06.2024

Rohini Palanisamy Sengottaian

Prashant Kataria

Gudimella Sridhar

Anupreksha Jain

Group 18

Table of Contents

Overview

This project aims to familiarize with generative text systems through two distinct tasks. In the first task, team will work with a clean, prepared dataset and gain hands-on experience fine-tuning a GPT model of their choice. While learning to fine-tune a GPT model for subject line generation, they will also create a new dataset for the next task. The trained QA model can then be deployed to test its ability to answer new AIML queries. This project offers a comprehensive learning experience, covering dataset curation, ideation, implementation, and deployment.

Goals

Fine-tune any GPT variant model for two tasks:

1. Generate a succinct subject line from the body of an email.
2. Model a system to generate an appropriate answer to a question related to AIML.

Project Objectives

Generate a succinct subject line from the body of an email.

Email Subject Line Generation task involves identifying the most important sentences in an email and abstracting their message into just a few words. The project provides an opportunity to work with generative models in NLP, specifically using GPT-2 variants, and to explore different metrics for evaluating text generation.

Model a system to generate an appropriate answer to a question related to AIML.

The main objective of “Model a system to generate an appropriate answer” is to develop a domain-specific GPT-variant model for answering AIML course questions. Pretrained models generally produce relevant text for open-domain prompts but often fail in domain-specific tasks. To address this, the model will be fine-tuned on a specialized dataset tailored for AIML-related questions. Participants will collaborate to create this dataset and, after fine-tuning, will evaluate the model's performance on unseen questions in the domain.

Literature Review

Generate a succinct subject line from the body of an email.

Existing Work:

Generating email subject lines is a specific instance of text summarization, a well-researched area in NLP. Traditional methods include extractive techniques, which select key sentences, and abstractive techniques, which generate new sentences. The advent of deep learning, particularly transformer-based models like BERT and T5, has significantly advanced the field.

Transformer Models: The Transformer architecture by Vaswani et al. (2017) revolutionized NLP by capturing long-range dependencies in text. Models like BERT and T5 have shown remarkable performance in text generation tasks.

GPT and BERT: These models can generate human-like text, making them strong candidates for generating email subject lines. The challenge is to ensure the subject lines are coherent, concise, and relevant to the email body.

Relevant Work:

The paper "This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation" by Zhang et al. (2021) specifically addresses the task of email subject line generation. The authors introduce a novel dataset and propose a model that combines both extractive and abstractive summarization techniques to generate subject lines. Their approach leverages a dual-attention mechanism to focus on the most relevant parts of the email body while generating a concise subject line. This work is particularly relevant as it provides a benchmark dataset and a baseline model that can be used for comparison.

Success Criteria:

1. Context Understanding: The model must understand the context and key points of the email body to generate a relevant subject line.
2. Conciseness: The subject line must be succinct while capturing the essence of the email.
3. Relevance: The generated subject line must be directly relevant to the email content to avoid misleading the recipient.

AIML Question Answering

Existing Work:

Question Answering (QA) systems have been a focal point of NLP research, with applications ranging from customer support to educational tools. Traditional QA systems relied on rule-based approaches and information retrieval techniques. However, the advent of deep learning has led to the development of more sophisticated models capable of understanding and generating natural language.

BERT and RoBERTa: Devlin et al. (2018) introduced BERT, a transformer-based model pre-trained on a large corpus of text in a bidirectional manner. BERT has been fine-tuned for various QA tasks, achieving state-of-the-art results. Liu et al. (2019) further improved upon BERT with RoBERTa, which optimized the pre-training process and achieved even better performance.

GPT-3: While BERT and RoBERTa are primarily used for understanding and classification tasks, GPT-3's generative capabilities make it suitable for generating answers to questions. Its ability to generate coherent and contextually relevant text has been demonstrated in various QA benchmarks.

Methodology

Approaches and Methods

Data Collection:

- Email Subject Generation: Collect a dataset of emails with their corresponding subject lines. This can be sourced from publicly available email datasets or generated synthetically.
  - The Annotated Enron Subject Line Corpus: https://github.com/ryanzhumich/AESLC
- AIML Question Answering: Compile a dataset of AIML-related questions and their corresponding answers.
  - This dataset will be created as a collective effort of all the teams participating in NLP projects as a part of the AIML course, and later used for fine tuning the GPT model.

- Each team will be provided with a question bank consisting of 250 questions each. The questions are to be provided with a short, 1-2 line answer to be entered in a CSV file.
- The given questions will be extracted from the course material already covered through the AIML lectures.

Model Selection:

Email Subject Generation

We will use the model highlighted in the paper, "This Email Could Save Your Life: Introducing the Task of Email Subject Line Genera@on", ACL 2019 and fine tune the model.

Team will be using multiple open source LLM models, i.e. llama, mistral and GPT, to train the datasets and generate summarization of the subject line.

Fine-Tuning:

1. Preprocessing: Clean and preprocess the datasets to ensure they are suitable for training. This includes tokenization, removing noise, and ensuring consistency.
2. Training: Fine-tune the model on the preprocessed datasets. Use techniques like transfer learning to leverage the pre-trained knowledge of the model.

AIML Question Answering:

Team will be trying both RAG based approach and LLM fine tuning for the task.

RAG-Based Approach:

For the AIML question answering task, we will employ a Retrieval-Augmented Generation (RAG) approach. RAG combines the strengths of retrieval-based and generation-based models to provide more accurate and contextually relevant answers.

1. Retriever: Use a retriever model (e.g., BM25, Dense Passage Retrieval) to fetch relevant documents or passages from a large corpus of AIML-related texts. This corpus can include a curated 250 Q & A dataset.
2. Generator: Use a generator model (like Mistral, Llama2/Llama3) to generate answers based on the retrieved documents. The generator will be fine-tuned to focus on the most relevant information from the retrieved documents.
3. Integration: Integrate the retriever and generator models to form a cohesive RAG system. The retriever fetches relevant documents, and the generator uses this information to generate accurate and contextually relevant answers.

LLM Fine-Tuning:

1. Preprocessing: Clean and preprocess the AIML question-answer dataset. Tokenize the text and ensure consistency in formatting.
2. Retriever Fine-Tuning: Fine-tune the retriever model on the AIML corpus to improve its ability to fetch relevant documents.
3. Generator Fine-Tuning: Fine-tune the generator model on the AIML question-answer dataset to improve its ability to generate accurate and relevant answers.
4. RAG System Training: Train the integrated RAG system to optimize the interaction between the retriever and generator models.

Tools and Frameworks:

- Libraries: Use libraries like Hugging Face's Transformers, PyTorch, and TensorFlow for model training and evaluation.
- Platforms: Utilize cloud platforms like AWS or Google Cloud for computational resources.
- Deployment platforms: FastAPI, Heroku, Ray Framework
- UI: Streamlit

Model Evaluation Criteria:

BLEU (Bi-Lingual Evaluation Understudy): This is a widely used metric for machine translation tasks. It measures the n-gram precision between the generated sequence and reference translations.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Similar to BLEU, ROUGE measures n-gram overlap but also considers recall. It comes in different variants like ROUGE-L (focuses on longest common subsequences) and ROUGE-N (focuses on n-grams).

WER (Word Error Rate): This metric simply calculates the percentage of words that are incorrect in the generated sequence compared to the reference.

CER (Character Error Rate): Similar to WER but focuses on individual characters instead of whole words.

Possible Outcomes in Stages

Stage 1: Data Collection and Preprocessing

- Successful collection and preprocessing of email and AIML question-answer datasets.

Stage 2: Model Fine-Tuning

- Fine-tuned custom model (https://arxiv.org/abs/1906.03497) for email subject generation.
- Fine-tuned LLM model for AIML question answering.

Stage 3: Evaluation and Optimization

- Evaluation of model performance using appropriate metrics.
- Optimization of models based on evaluation results.

Stage 4: Deployment and Testing

- Deployment of the models as APIs for real-world testing.
- Collection of user feedback and further refinement of models.

Challenges and Risks

Computation

- Training of model requires GPU and more computational power
- The amount of computation and GPU/CPU availability can influence the Model selection and tuning of it.

Contextual data availability and Quality of Answers

- Collecting enough contextual data will be a challenge as LLM models are data hungry and require more data to learn
- Quality of data can not be evaluated manually or automatically before training which might affect overall Q&A output of the model.

- The quality of accuracy and relavance of the answers are directly impacted by the data that will be used.

Timelines

Date	Submission
29th Jun	Project Proposal
6th July	Preliminary Model & Training (on subset of data)
2nd August	Model building, training for complete dataset
17th August	Deployment testing (on subset of data)
1st September	Fine tuning & Final deployment
15th September	Final Presentation

Applicability in the Real World

Email Subject Generation:

- Business Use: Automate the generation of email subject lines to improve email marketing campaigns and internal communications.
- Personal Use: Assist individuals in crafting effective subject lines for personal and professional emails.

AIML Question Answering:

- Educational Use: Provide accurate and instant answers to students' questions related to AIML, enhancing the learning experience.
- Professional Use: Assist professionals in quickly finding answers to technical questions, improving productivity and knowledge sharing.

Conclusion

This project will provide comprehensive exposure to generative text systems, covering both fine-tuning and deployment aspects. By completing these tasks, the team will gain valuable experience in working with state-of-the-art machine learning models and contribute to developing practical AI applications.