**Capstone Project Report**

**Title: Email Subject Line Generation and AIML Question Answering System**

**Introduction**

This capstone project focuses on two primary tasks:

1. Generating succinct subject lines from email bodies
   1. In this task, team worked on preprocessing the dataset and fine tune multiple llm models and picked the llm model with good metrics scores by using evaluating techniques.
2. Developing a system to answer questions related to AIML (Artificial Intelligence Markup Language).
   1. In this task, team worked on preparing the dataset and preprocessing the dataset. and fine tune multiple llm models and picked the llm model with good metrics scores by using evaluating techniques.
3. The project also involves in deploying the fine tuned models using Hugging Face Spaces with Gradio and FastAPI.

**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

**Methodology**

**Task 1: Email Subject Line Generation**

1. **Data Collection**:
   * The Annotated Enron Subject Line Corpus: <https://github.com/ryanzhumich/AESLC> dataset is used.
   * Preprocessed the data to remove punctuation marks, lower case the text for consistency to ensure quality and consistency.
2. **Model Selection and Fine-Tuning**:
   * Chosen 2 generative llm models and 2 transformer-based models developed for natural language processing tasks for fine tuning as listed in table below

|  |  |  |  |
| --- | --- | --- | --- |
| LLM | Framework | Model Type | Training Steps |
| Mistral | unsloth | 4 bit quantized | 60 |
| Llama3 | unsloth | 4 bit quantized | 60 |
| T5 | Transformer | Base model | 200 |
| Bart | Transformer | Base model | 200 |

* + We had following observations
    1. Generative llm models are very large to be trained on base model, we had to use quantized versions. Also for training we used [PEFT](https://huggingface.co/docs/peft/en/package_reference/lora)
    2. Email subjects generated by both Generative and Seq2Seq Model is contextually correct.
    3. Generative llm models generate word's synonyms, hence Rogue score is very low.
    4. Seq2Seq models pick up the words from Email content, which is expected in this experiment, hence Rouge score is high.
  + From the above observations, we picked Bart model as it is a Seq2Seq model and better suited for the task in hand and giving better metrics.

1. **Training:**
   * **Mistral**
   * **Code File**: [Group18EmailDataSetTrainingMistral.ipynb](https://github.com/anukvma/group18_email_subject_generation/blob/main/Group18EmailDataSetTrainingMistral.ipynb)  
     **Model**: unsloth/mistral-7b-v0.3-bnb-4bit

|  |
| --- |
| *FastLanguageModel.get\_peft\_model(*  *model,*  *r = 16, # Choose any number > 0 ! Suggested 8, 16, 32, 64, 128*  *target\_modules = ["q\_proj", "k\_proj", "v\_proj", "o\_proj",*  *"gate\_proj", "up\_proj", "down\_proj",],*  *lora\_alpha = 16,*  *lora\_dropout = 0, # Supports any, but = 0 is optimized*  *bias = "none", # Supports any, but = "none" is optimized*  *# [NEW] "unsloth" uses 30% less VRAM, fits 2x larger batch sizes!*  *use\_gradient\_checkpointing = "unsloth", # True or "unsloth" for very long context*  *random\_state = 3407,*  *use\_rslora = False, # We support rank stabilized LoRA*  *loftq\_config = None, # And LoftQ*  *)*  *Training Framework: Huggingface trl [SFTrainer](https://huggingface.co/docs/trl/v0.9.6/en/sft_trainer" \l "trl.SFTTrainer) Training Arguments:*  *TrainingArguments(*  *per\_device\_train\_batch\_size = 2,*  *per\_device\_eval\_batch\_size=2,*  *gradient\_accumulation\_steps = 4,*  *evaluation\_strategy="steps",*  *warmup\_steps = 5,*  *num\_train\_epochs=3,*  *max\_steps = 60, # Set num\_train\_epochs = 1 for full training runs*  *learning\_rate = 2e-4,*  *fp16 = not is\_bfloat16\_supported(),*  *bf16 = is\_bfloat16\_supported(),*  *logging\_steps = 1,*  *optim = "adamw\_8bit",*  *weight\_decay = 0.01,*  *lr\_scheduler\_type = "linear",*  *seed = 3407,*  *output\_dir = "outputs",*  *# report\_to="wandb", # enable logging to W&B*  *logging\_strategy = 'steps',*  *# save\_total\_limit=2,*  *)* |

* + LLAMA3
  + **Code File**: [Group18FineTuneLlama3EmailSubjectFinal.ipynb.ipynb](https://github.com/anukvma/group18_email_subject_generation/blob/main/Group18FineTuneLlama3EmailSubjectFinal.ipynb.ipynb)  
    Model: unsloth/llama-3-8b-bnb-4bit

|  |
| --- |
| *FastLanguageModel.get\_peft\_model(*  *model,*  *r = 16, # Choose any number > 0 ! Suggested 8, 16, 32, 64, 128*  *target\_modules = ["q\_proj", "k\_proj", "v\_proj", "o\_proj",*  *"gate\_proj", "up\_proj", "down\_proj",],*  *lora\_alpha = 16,*  *lora\_dropout = 0, # Supports any, but = 0 is optimized*  *bias = "none", # Supports any, but = "none" is optimized*  *# [NEW] "unsloth" uses 30% less VRAM, fits 2x larger batch sizes!*  *use\_gradient\_checkpointing = "unsloth", # True or "unsloth" for very long context*  *random\_state = 3407,*  *use\_rslora = False, # We support rank stabilized LoRA*  *loftq\_config = None, # And LoftQ*  *)*  *Training Framework: Huggingface trl [SFTrainer](https://huggingface.co/docs/trl/v0.9.6/en/sft_trainer" \l "trl.SFTTrainer) Training Arguments:*  *TrainingArguments(*  *per\_device\_train\_batch\_size = 2,*  *gradient\_accumulation\_steps = 4,*  *warmup\_steps = 5,*  *max\_steps = 60,*  *learning\_rate = 2e-4,*  *fp16 = not is\_bfloat16\_supported(),*  *bf16 = is\_bfloat16\_supported(),*  *logging\_steps = 1,*  *optim = "adamw\_8bit",*  *weight\_decay = 0.01,*  *lr\_scheduler\_type = "linear",*  *seed = 3407,*  *output\_dir = "outputs",*  *)* |

* + T5
  + **Code File**: [Group18FineTuningT5EmailSubject.ipynb](https://github.com/anukvma/group18_email_subject_generation/blob/main/Group18FineTuningT5EmailSubject.ipynb)  
    Model: t5-base  
    Training Framework: Transformer Seq2SeqTrainer

Training Arguments:

|  |
| --- |
| *Seq2SeqTrainingArguments(*  *model\_dir,*  *evaluation\_strategy="steps",*  *eval\_steps=200,*  *logging\_strategy="steps",*  *logging\_steps=100,*  *save\_strategy="steps",*  *save\_steps=200,*  *learning\_rate=4e-5,*  *per\_device\_train\_batch\_size=8,*  *per\_device\_eval\_batch\_size=8,*  *weight\_decay=0.01,*  *save\_total\_limit=3,*  *num\_train\_epochs=2,*  *predict\_with\_generate=True,*  *fp16=True,*  *load\_best\_model\_at\_end=True,*  *metric\_for\_best\_model="rouge1",*  *report\_to="tensorboard"*  *)* |

* + **Bart**
    1. **Code File**: [Group18FineTuneBartEmailSubjectFinal.ipynb](https://github.com/anukvma/group18_email_subject_generation/blob/main/Group18FineTuneBartEmailSubjectFinal.ipynb)
    2. **Model:** facebook/bart-large-xsum
    3. Training Framework: Transformer Seq2SeqTrainer
    4. Training Arguments:

|  |
| --- |
| *Seq2SeqTrainingArguments(*  *model\_dir,*  *evaluation\_strategy="steps",*  *eval\_steps=200,*  *logging\_strategy="steps",*  *logging\_steps=100,*  *save\_strategy="steps",*  *save\_steps=200,*  *learning\_rate=4e-5,*  *per\_device\_train\_batch\_size=8,*  *per\_device\_eval\_batch\_size=8,*  *weight\_decay=0.01,*  *save\_total\_limit=3,*  *num\_train\_epochs=2,*  *predict\_with\_generate=True,*  *fp16=True,*  *load\_best\_model\_at\_end=True,*  *metric\_for\_best\_model="rouge1",*  *report\_to="tensorboard"*  *)* |

1. **Results:**
2. **Evaluation**:
   * For the evaluation of models used ROUGE metrics to measure the quality of generated subject lines.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LLM | Rogue1 | Rogue2 | RougeL | RogueLSum |
| Mistral | 0.04 | 0.02 | 0.04 | 0.04 |
| Llama3 | 0.04 | 0.02 | 0.04 | 0.04 |
| T5 | 0.14 | 0.07 | 0.14 | 0.14 |
| Bart | 0.27 | 0.13 | 0.25 | 0.25 |

1. **Deployment**:
   * Deployed the models using Gradio for an interactive web interface.
     1. <https://huggingface.co/spaces/GSridhar1982/EmailSubjectGenerationDemo>
   * Implement FastAPI for API access.
     1. [https://anukvma-emailsubjectapi.hf.space](https://anukvma-emailsubjectapi.hf.space/)  
        Code files: [api(folder)](https://github.com/anukvma/group18_email_subject_generation/tree/main/api): Code for Fast API includes Dockerfile, requirements.txt and main.py
     2. Curl command

|  |
| --- |
| *curl --location --request GET 'https://anukvma-emailsubjectapi.hf.space' \*  *--header 'Content-Type: application/json' \*  *--data-raw '{*  *"model\_name":"anukvma/bart-base-medium-email-subject-generation-v5",*  *"email\_content": "Harry - I got kicked out of the system, so I'\''m sending this from Tom'\''s account. He can fill you in on the potential deal with STEAG. I left my resume on your chair. I'\''ll e-mail a copy when I have my home account running. My contact info is:"*  *}'* |

**Task 2: AIML Question Answering**

1. **Data Collection and Preprocessing**:
   * Compiled a dataset of AIML-related questions and answers from the AIML course materials.
   * Ensure the dataset covers all units within AIML course.
   * Preprocessed the data to remove punctuation marks, lower case the text for consistency to ensure quality and consistency.
2. **Model Selection and Fine-Tuning**:
   * Chosen 3 generative llm models and 1 transformer-based models for fine tuning as listed in table below

|  |  |  |  |
| --- | --- | --- | --- |
| LLM | Framework | Model Type | Training Steps |
| Mistral | unsloth | 4 bit quantized | 60 |
| Llama3.1 | unsloth | 4 bit quantized | 60 |
| GPT2 | Transformer (lora) | Base model | 600 |
| Bart | Transformer | Base model | 600 |

* + We had following observations
    1. Generative llm models are very large to be trained on base model, we had to use quantized versions. Also for training we used [PEFT](https://huggingface.co/docs/peft/en/package_reference/lora)
    2. Answers generated by all models are correct.
    3. GPT2 is giving better metrics compared to other models
  + Used PEFT techniques primarily LoRA to fine-tune the llm models on the AIML dataset for question answering.

1. **Evaluation**:
   * Assess the model's performance using ROUGE metrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LLM | Rogue1 | Rogue2 | RougeL | RogueLSum |
| Bart | 0.383593 | 0.17502 | 0.313383 | 0.405906 |
| GPT2 | 0.482330 | 0.177874 | 0.394122 | 0.427295 |
| Llama31 | 0.20 | 0.10 | 0.22 | 0.23 |
| Mistral7b | 0.26 | 0.08 | 0.21 | 0.20 |

1. **Deployment**:
   * Used Gradio to create a user-friendly interface for question answering.
   * **Gradio App:** [anukvma/Question\_Answer](https://huggingface.co/spaces/anukvma/Question_Answer)  
     **Gradio App:** [GSridhar1982/QA\_Llama31\_FineTuned] ([<https://huggingface.co/spaces/GSridhar1982/QA_Llama31_FineTuned>])
   * Set up FastAPI for backend API services.

**FAST API:**[**hugging face space**](https://huggingface.co/spaces/anukvma/AIMLQnAAPI) **Code files: [api(folder)](https://github.com/anukvma/group18_final_project/tree/main/aiml_question_answers/api):** Code for Fast API includes Dockerfile, requirements.txt and main.py  
Curl command for API call:

|  |
| --- |
| *curl --location --request GET 'https://anukvma-aimlqnaapi.hf.space' \*  *--header 'Content-Type: application/json' \*  *--data-raw '{*  *"question": "what is linear regression?"*  *}'*  **Response**: "Linear regression is a statistical method used to forecast the probability of a dependent variable using a linear equation." |

**Implementation**

**Tools and Technologies**

* **Hugging Face Transformers**: For accessing and fine-tuning pre-trained models.
* **Gradio**: To build interactive web applications for model deployment.
* **FastAPI**: For creating robust APIs.
* **Unsloth**: Framework for training the large language models
* **Python**: The primary programming language for model development and deployment.

**Steps**

1. **Data Preprocessing**:
   * Clean and tokenize the datasets.
   * Split data into training, validation, and test sets.
2. **Model Fine-Tuning**:
   * Use transfer learning to adapt pre-trained models to specific tasks.
   * Optimize hyperparameters for improved performance.
3. **Deployment**:
   * Develop Gradio interfaces for user interaction.
   * Implement FastAPI endpoints for seamless integration.

**Results**

* **Email Subject Line Generation**:
  + Achieved a ROUGE score of 0.25 using a BART model, indicating effective subject line generation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LLM | Rogue1 | Rogue2 | RougeL | RogueLSum |
| Mistral | 0.04 | 0.02 | 0.04 | 0.04 |
| Llama3 | 0.04 | 0.02 | 0.04 | 0.04 |
| T5 | 0.14 | 0.07 | 0.14 | 0.14 |
| **Bart** | **0.27** | **0.13** | **0.25** | **0.25** |

* **AIML Question Answering**:
  + Achieved a ROUGE score of 0.39 using a GPT2 model, demonstrating reliable question answering capabilities.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LLM | Rogue1 | Rogue2 | RougeL | RogueLSum |
| Bart | 0.383593 | 0.17502 | 0.313383 | 0.405906 |
| **GPT2** | **0.482330** | **0.177874** | **0.394122** | **0.427295** |
| Llama31 | 0.20 | 0.10 | 0.22 | 0.23 |
| Mistral7b | 0.26 | 0.08 | 0.21 | 0.20 |

**Conclusion**

The project successfully developed and deployed systems for generating email subject lines and answering AIML-related questions. The use of LLMs and deployment tools like Gradio and FastAPI facilitated efficient model training and user interaction.

**Future Work**

* Enhance model performance with larger datasets and more advanced architectures.
* Expand the question answering system to cover additional AI topics.
* Integrate feedback mechanisms to continuously improve model accuracy.

**References**

* Hugging Face Transformers Documentation
* Unsloth Documentation
* Gradio Documentation
* FastAPI Documentation