PGCP - Capstone Project

(Email Subject Line Generation &
Question Answering on AIML Queries)

AIML-B22-Group-15:

Shilpa Sirikonda

Sudheendra Gopinath

Ravi Kanth Jami

Lohith Reddy Manchireddy

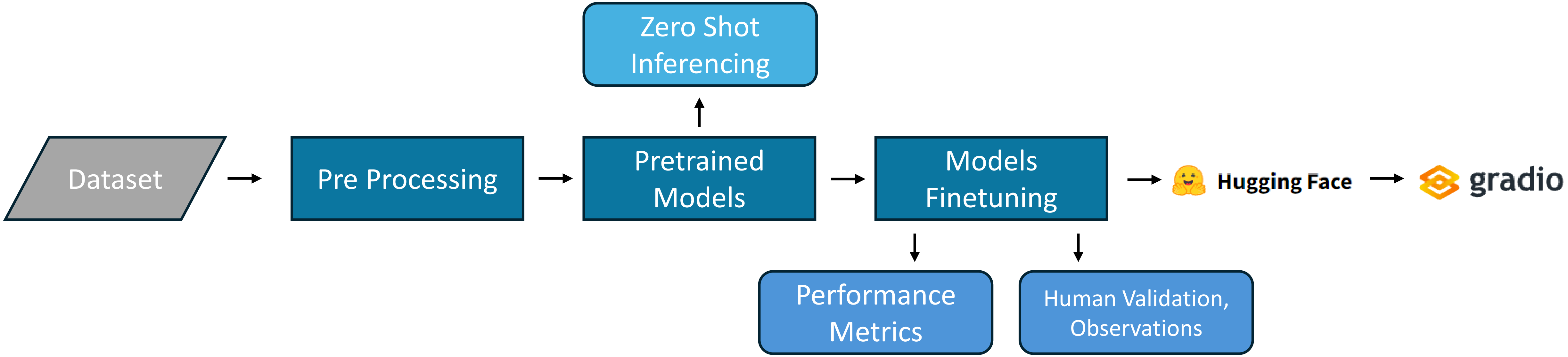
Project Objective

To familiarize participants with generative text systems through two distinct tasks i.e., to fine-tune suitable GPT variant models for each of two tasks:

Generate a succinct subject line for a given email body

Generate an answer for a given question related to AIML

Workflow



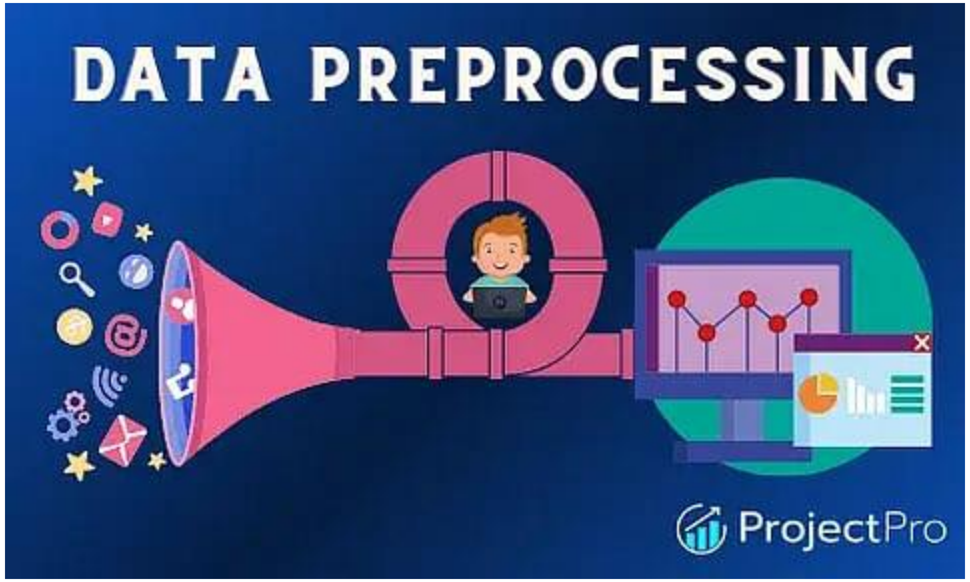
Task: Fine-tune few summarization GPT model to generate concise email subject lines and deployment.

Understanding Dataset

Parameter	Quantity
# of training samples	14,436
# of validation samples	1,960
# of test samples	1,906
Average length of Emails	75 words
# of human annotations per Email (test dataset)	3
Average length of human annotations	4 words

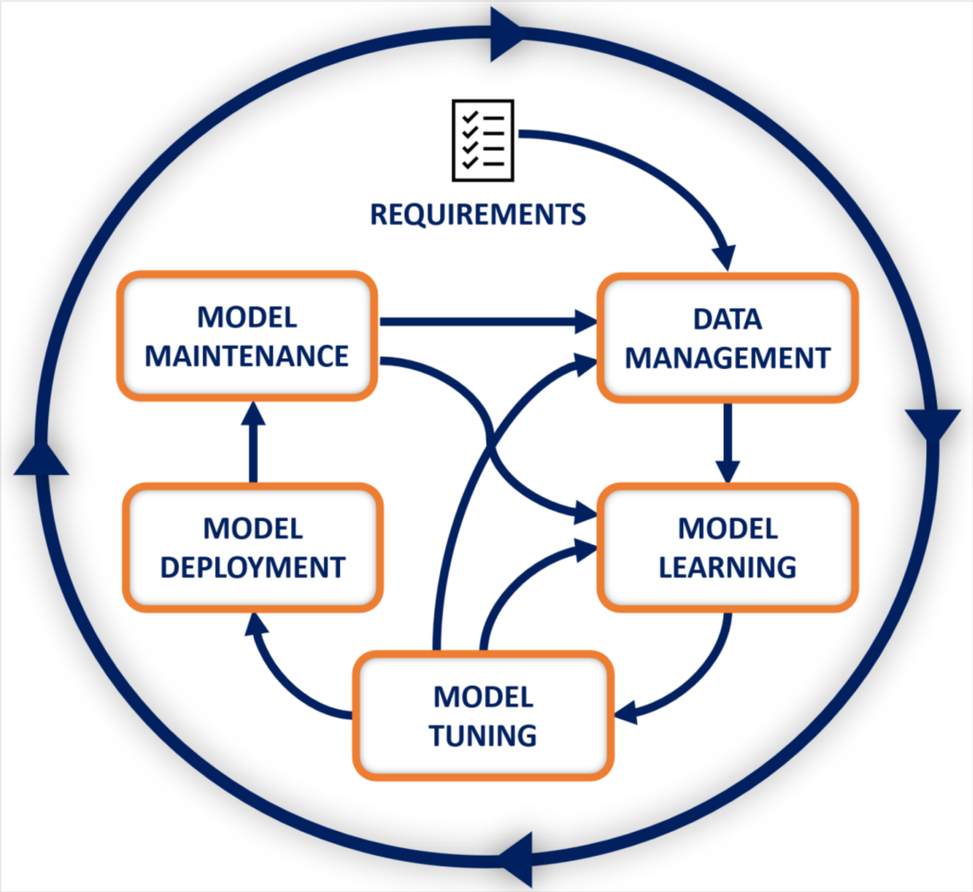
Dataset : Annotated Enron Subject Line Corpus

Data Loading and Pre-Processing



- **LangChain_community.DirectoryLoaders** are used to load the email files and then converted to Pandas DataFrame. (LangChain document_loader was found to be **organized, scalable and easy to use**)
- Evaluation (dev, test) split of the data contains 3 annotated subject lines by human annotators. Multiple possible references facilitate a better evaluation of the generated subject, since it is difficult to have only one unique, appropriate subject per email
- A subset of train dataset is created for finetuning language models, although full train data set is also used.

Methodology



- On high level different open source language models are researched and assessed that suits the problem statement of extracting most important words/ context/ concise summarization.
- **Transformer** models and **Bart** models were found to be most apt for the given task other than the ChatGPT models.
- Couple of pretrained models were selected to test with zero-shot inferencing and further finetuning
- Utilize built-in Hugging Face/ SFTTrainer Trainer class for training
- Check performance of models through Qualitative and Quantitative evaluation
- Model Deployment

Finetuning the models

Model	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
FLAN-T5	0.3189	0.1852	0.3108	0.3100
facebook/bart-base	0.2882	0.1232	0.2879	0.2893
google-gemma-with-unsloth	**0.6355**	**0.4207**	**0.5924**	**0.5924**
unsloth/ mistral-7b-v0.3-bnb-4bit	0.2235	0.715	0.2236	0.2262
unsloth/ Phi-3-mini-4k-instruct-bnb-4bit	0.1063	0.0250	0.0942	0.0946

Finetuned Models: Performance Metrics

Prompt Illustration examples

<table>
<tr><th>Model</th><th colspan="2">Prompt</th></tr>
<tr><td>FLAN-T5</td><td colspan="2">Email-Subject (prompt-input-response) format is created as explicit instructions for the LLM. Prepend a prompt instruction to the start of email body and generate the subject with Suject as follows:

Training prompt (email):

prompt = f""" Generate a subject line for the following email.

Email: {email}

Subject:
"""</td></tr>
<tr><td>google-gemma-with-unsloth</td><td>instruction = "Generate a subject line for the following email."

if x['body']:
formatted_text = f"""Below is an instruction that describes a task. \
Write a response that appropriately completes the request.

Instruction:
{instruction}

Input:
{x['body']}

Response:
{x['subject']}"""</td><td>Instruction = “Generate a subject line for the following email.”

If x[‘body’]:
formatted_text = f”””Below is an instruction that describes a task. \
Write a response that appropriately completes the request.

Instruction:
{instruction}

Input:
{x[‘body’]}

Response:
“””</td></tr>
</table>

Finetuned model- Human Evaluation

*Google's Flan-T5:

Out[99]:

	original_subjects	T5_subjects	finetuned_flan_T5_subjects
0	DPL	CSFB	CSFB
1	Toronto Dominion (Texas), Inc. (EXISTING ISDA ...	Toronto-Dominion Bank	Toronto-Dominion Bank
2	PLEASE APPROVE - 3 PLASTIC PRODUCTS - URGENT	Product Type approval	Product Type Approval
3	Update on Schedule	Wes Colwell's PRC Meeting	Wes Colwell's PRC Meeting
4	Cargill, Incorporated	Cargill	Cargill
5	Forum for Solution to Utility Undercollection ...	PUC's Uncollection Problem	PUC's Uncollection Problem
6	We want your business!! 5.85% Fixed Rate	IMPORTANT - CHANGE OF TIME FOR CHANGE!	IMPORTANT - CHANGE OF TIME FOR CHANGE!
7	PG&E Prior Guarantees	Guaranty	Guaranty
8	IntercontinentalExchange - Update	Aventail Login	Aventail Login

*Facebook's Bart-Base:

	True_subjects	FB_Bart_subjects	Finetuned_FB_Bart_subjects
0	Enron Situation and Technology For All: A Note...	Technology for All	Technology for All
1	Organizational Announcement	Organizational Announcement	Organizational Announcement
2	Your approval is requested	MIGHOOD PRIVILEGED PRIVILEGED PRIVILEGED PRIVI...	MVP Request
3	Securities Loan Agreement	Enron Credit Inc.	Enron Credit Inc.
4	Request for Deal #	QQ6739.2	PG&E
5	Boxes	Boxes	Boxes
6	Volume Management Addition: Eugene Lee	Employee Announcement: Eugene Lee	Employee Announcement
7	Master Agreement	C constellation power source	C constellation power source
8	Charles' Christening	Meeting with charles	Meeting with charles

Finetuned model- Human Evaluation

*Gemma 7B:

	True_subjects	gemma_subjects	ann0	ann1	ann2
0	CALIFORNIA UPDATE - 9/4/2001	Edison MOU	executive summary	executive summary: edison mou	update on appropriations committee's plan
1	PLEASE READ - IMPORTANT INFORMATION FOR PARTIC...	Enron Savings Plan Changes	please note this amendment to the enron saving...	important changes made to enron corp. savings ...	important amendments to enron corp. savings plan
2	Franky Sulistio	Franky Sulistio	fran sulistio introduction	new addition to gas fundamentals it group: fra...	franky sulistio joining gas fundamentals it group
3	Turlock Irrigation District	Master Firm Purchase /Sale Agreement for Turlock	here's the turlock agreement draft to distribute	turlock firm purchase/sale agreement instructions	master firm purchase/ sale agreement draft
4	NDA-Sabre Corporation	NDA for EBS	david endicott's review needed for non-disclos...	proposed non-disclosure agreement for review	comments please: nda draft + changes
5	high volume trading counterparties	ISDA Master Agreements	preparing isda master agreements	upgrade and update master agreements	master agreements update

Evaluate the models quantitatively (with Rouge)

```
Prediction: Proposals for the CES meeting
Reference: Options the Governor's Considering
ROUGE Scores: {'rouge1': 20.000000000000004, 'rouge2': 0.0, 'rougeL': 20.000000000000004}

Prediction: NDA - Enron North America Corp.
Reference: NDA - IntercontinentalExchange
ROUGE Scores: {'rouge1': 28.571428571428577, 'rouge2': 0.0, 'rougeL': 28.571428571428577}

Prediction: Gas Turbine/Condition Monitoring Conference and Workshop
Reference: Conference & workshop
ROUGE Scores: {'rouge1': 44.44444444444445, 'rouge2': 0.0, 'rougeL': 44.44444444444445}

Prediction: ECCO docs
Reference: Delta docs
ROUGE Scores: {'rouge1': 50.0, 'rouge2': 0.0, 'rougeL': 50.0}
```

Sample: Individual record wise.
Scores in %

Evaluate the models quantitatively (with Rouge)

```
instruct_model_results = rouge.compute(
    predictions=trained_facebook_bart_subjects,
    references=human_annoted_subjects[0:len(trained_facebook_bart_subjects)],
    use_aggregator=True,
    use_stemmer=True,
)

print('ORIGINAL MODEL:')
print(original_model_results)
print('INSTRUCT MODEL:')
print(instruct_model_results)
```

```
ORIGINAL MODEL:
{'rouge1': 0.06252371479992963, 'rouge2': 0.014090909090909, 'rougeL': 0.06187822558905744, 'rougeLsum': 0.061153489805264
84}
INSTRUCT MODEL:
{'rouge1': 0.365, 'rouge2': 0.13571428571428573, 'rougeL': 0.365, 'rougeLsum': 0.3716666666666667}
```

Scores: Facebook Bart Average wise

Evaluate the models quantitatively (with Rouge)

	rouge1	rouge2	rougeL	rougeLsum
True_subjects Vs. Base_model	0.238393	0.120889	0.234841	0.232768
True_subjects Vs. Finetuned_model	0.265899	0.106019	0.258892	0.260335
Ann0 Vs. Base_model	0.265871	0.145368	0.248917	0.249096
Ann0 Vs. Finetuned_model	0.292435	0.138663	0.274334	0.275530
Ann1 Vs. Base_model	0.272054	0.145496	0.251515	0.251896
Ann1 Vs. Finetuned_model	0.307122	0.190622	0.300536	0.301829
Ann2 Vs. Base_model	0.250739	0.134479	0.240002	0.238884
Ann2 Vs. Finetuned_model	0.318985	0.185258	0.310836	0.310023

Flan T5 Rouge Scores w.r.t. original subjects and three other human annotations

Absolute percentage improvement of FINETUNED MODEL over PRETRAINED

For FLAN T5 : Absolute percentage improvement of FINETUNED MODEL over PRETRAINED

```
print("For Ture Subjects: Absolute percentage improvement of FINETUNED MODEL over PRETRAINED")

improvement = (np.array(list(finetuned_model_results.values())) - np.array(list(original_model_r
for key, value in zip(finetuned_model_results.keys(), improvement):
    print(f'{key}: {value*100:.2f}%')
```

```
For Ture Subjects: Absolute percentage improvement of FINETUNED MODEL over PRETRAINED
rouge1: 2.75%
rouge2: -1.49%
rougeL: 2.41%
rougeLsum: 2.76%
```

FB Bart

```
print("For Ann2 : Absolute percentage improvement of FINETUNED MODEL over PRETRAINED")

improvement = (np.array(list(ann2_finetuned_model_results.values())) - np.array(list(ann2_origin
for key, value in zip(finetuned_model_results.keys(), improvement):
    print(f'{key}: {value*100:.2f}%')
```

```
For Ann2 : Absolute percentage improvement of FINETUNED MODEL over PRETRAINED
rouge1: 6.82%
rouge2: 5.08%
rougeL: 7.08%
rougeLsum: 7.11%
```

Evaluate the models quantitatively (with Rouge)

```
Gemma_Subjects Vs. Original Subjects:
rouge1: Precision=0.3612, Recall=0.2909, F1=0.2776
rouge2: Precision=0.2421, Recall=0.1630, F1=0.1563
rougeL: Precision=0.3612, Recall=0.2909, F1=0.2776
rougeLsum: Precision=0.3612, Recall=0.2909, F1=0.2776
```


```
Ann0 Vs. Gemma_model:
rouge1: Precision=0.5369, Recall=0.3044, F1=0.3621
rouge2: Precision=0.2937, Recall=0.1318, F1=0.1659
rougeL: Precision=0.4978, Recall=0.2791, F1=0.3335
rougeLsum: Precision=0.4978, Recall=0.2791, F1=0.3335

Ann1 Vs. Gemma_model:
rouge1: Precision=0.6355, Recall=0.3667, F1=0.4326
rouge2: Precision=0.4207, Recall=0.2094, F1=0.2628
rougeL: Precision=0.5924, Recall=0.3260, F1=0.3950
rougeLsum: Precision=0.5924, Recall=0.3260, F1=0.3950

Ann2 Vs. Gemma_model:
rouge1: Precision=0.5366, Recall=0.3721, F1=0.4060
rouge2: Precision=0.3214, Recall=0.1849, F1=0.2190
rougeL: Precision=0.5106, Recall=0.3427, F1=0.3802
rougeLsum: Precision=0.5106, Recall=0.3427, F1=0.3802
```

Gemma_7b_with_Unsloth Scores w.r.t. original subjects and three other human annotations

Key Arguments for gemma-7b TrainingArguments

- **per_device_train_batch_size**: It is set to 1, meaning 1 examples will be processed per device in each step.
- **gradient_accumulation_steps:** Grad Accumulation steps before performing a parameter update. Increases the batch size by accumulating gradients over multiple steps. Here, it is set to 2, meaning gradients will be accumulated over 2 steps before updating the model parameters.
- **warmup_steps**: This sets the number of warm-up steps during training, gradually increasing the Learning Rate from 0 to the provided value. We set to 5, so the Learning Rate will linearly increase over the first 5 steps.
- **max_steps:** Total number of training steps to perform. Here, it is set to 50, meaning the training will stop after 50 steps.
- **learning_rate**: First Learning Rate used for training. We set it to 2e-4
- **fp16 and bf16**: Control the precision used for training. fp16 is for half-precision (16-bit) training, while bf16 is for bfloat16 training if GPU supported.
- **logging_steps**: Sets the interval at which training metrics and losses are logged. We set it to 1, so logs are printed after every training step.
- **optim:** Optimizer to use for training. We set it to 'paged_adamw_8bit', a specialized optimizer for memory-efficient training.
- **weight_decay**: Weight Decay Rate that we need for regularization. Set to 0.01.
- **lr_scheduler_type:** Learning Rate Scheduler to use during training, "linear“.

Feature/ Model	FLAN-T5 (Fine-tuned Language-Agnostic T5)	Facebook BART (Bidirectional and Auto-Regressive Transformers)	Gemma 7B Unsloth	Mistral	Phi-3
Developer	Google	Facebook AI (Meta AI)	Google	Mistral AI	Microsoft
Architecture	Encoder-Decoder (Transformer)	Encoder-Decoder (Transformer)	Transformer-based	Transformer-based	LLM (Transformer-based)
Pre-training Objective	Span-based masked language modeling (MLM), fine-tuned on diverse tasks	Denoising autoencoder (corrupted input reconstruction)	Autoregressive, optimized for unslothing tasks	Autoregressive/Bidirectional	Scalable alignment with focus on safety and ethics
Key Feature	Fine-tuned for zero-shot and few-shot learning across multiple languages	Combines bidirectional encoding with autoregressive decoding	Specialized for unslothing tasks	Optimized for specific domains	Designed for ethical AI and human alignment
Embeddings/ Positional Encodings	Learnable token embeddings. Relative positional encodings	Learnable token embeddings . Absolute positional encodings	Learnable token embeddings. Relative or absolute positional encodings	Learnable token embeddings. Relative or absolute positional encodings	Custom relative or absolute positional encodings depending on task alignment
Tokenizer	SentencePiece, byte-pair encoding (BPE)	Byte-Pair Encoding (BPE)	Byte-Pair Encoding (BPE)	Grouped-Query Attention, Sliding-Window Attention,Byte-fallback BPE tokenizer	Custom tokenizer designed for alignment with human preferences
Model Size/ Parameters	Multiple sizes (Small to XXL: 60M to 11B parameters)	Multiple sizes (Base to Large: 139M to 406M parameters)	7 billion parameters	Multiple sizes expected. Range from 100M to 7B+ depending on configuration	Large-scale models - 3.8 billion parameter to 10B+ parameters in larger models with scalable versions
Applications	Text generation, translation, summarization, classification, few-shot learning, and more	Text generation, summarization, translation, and sequence-to-sequence tasks	Specialized NLP applications (Unslothing tasks)	Domain-specific NLP applications	Broad NLP tasks large reasoning capabilities or are highly specialized (Synthetic Text Generation, Code Generation, RAG, or Agents).

Email Subject -Observations/ Key Learnings

- Fine-tuned models show performance improvement. They are effectively capturing key points and overall essence, with improved ROUGE-1 scores showing alignment with essential topics.
- The models demonstrated potential for **understanding nuanced details**, as indicated by ROUGE-2 scores, though there is room for improvement.
- Higher ROUGE-L and ROUGE-Lsum scores reflect good maintenance of subject length and relevance.
- Specific prompts, such as "generate a subject line" yield better results compared to combined prompts like "summarize the text“.
- Repetitive responses in pre-trained models (e.g., Mistral) are managed by applying a repetition_penalty of 1.5, but excessive penalties cause unusual outputs.
- Phi3 excels in text completion and GPT-style conversations but may produce hallucinations and less accurate results.

DEPLOYMENT - Building App with Gradio and publishing in Hugging Face

gradio

Hugging Face

- **Build the Gradio App**: Designed Gradio interface, defining how the user will interact with the model and ensuring the input and output specifications are clear.
- **Save the App and Dependencies**: Prepared our app script and ensure all necessary dependencies are listed in a requirements file, ready for deployment.
- **Publish on Hugging Face Spaces**: Created an account on Hugging Face, set up a new Space for our app, and push our code to this Space, making our app publicly accessible.

TASK-2

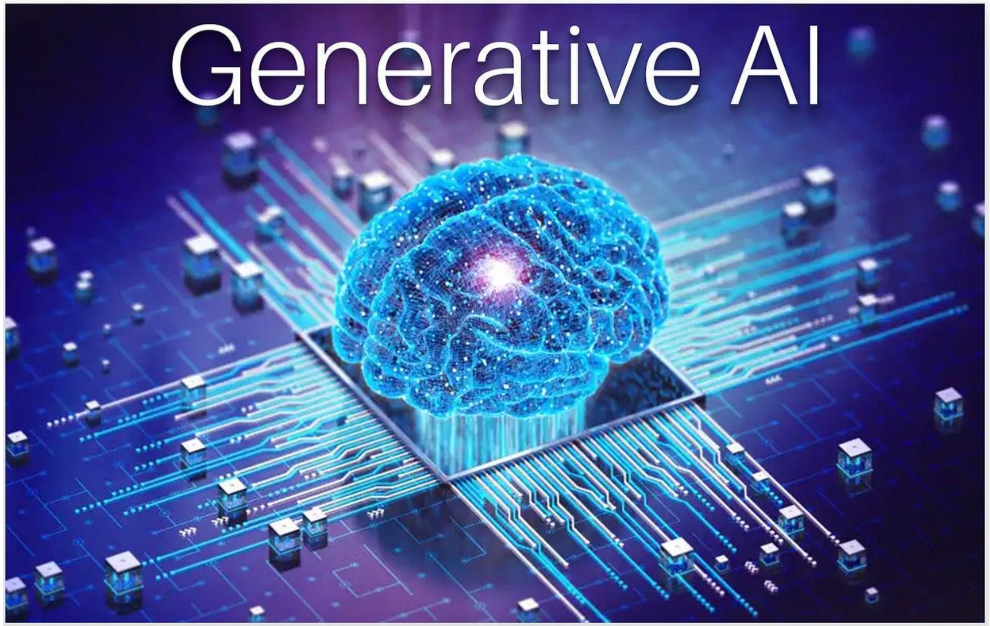
Generate an answer for a given question related to AIML

Understanding QA Dataset

Parameter	Quantity
# of training samples	1,300
# of validation samples	80
# of test samples	120
Average length of questions	10 words
# of human answers per Question (test dataset)	2
Average length of human annotations	50 words

- Dataset-1 has a question-and-answer pair for train set and, a question and two human annotated answers for the test and dev sets.
  - Train set -(1316, 2)
  - Test set (120, 3)
  - Dev set (80, 3)
- Dataset-2 has a question-and-answer pair
  - Train set -(1985, 2)
  - Test set -(249, 2)
  - Dev set - (248, 2))

Models Approaches



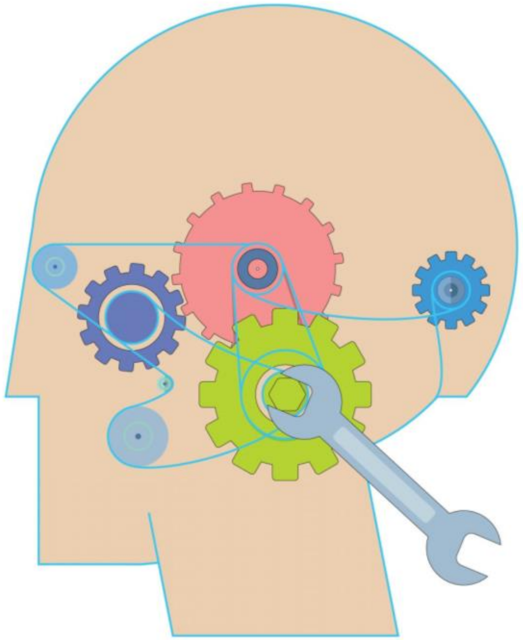
- Both **extractive** and **sequence-to-sequence** approaches were explored for the given problem statement
- Sequence-to-Sequence (Seq2Seq) approach was selected for the task due to following reasons:
  - AI/ML knowledge corpus usage was not recommended to use
  - Flexible Output Generation: Seq2Seq models generate new sequences, unlike extractive models, which are restricted to selecting text spans, making them ideal for tasks like summarization or translation.

GPT Models' selection for finetuning

Models' comparison

Feature	Gemma 7B	LLaMA 3 8B	GPT-2
Model Size	7 billion parameters	8 billion parameters	1.5 billion parameters
Training Data	Task-specific, proprietary data	General-purpose, large public datasets	Large-scale, web-based text data
Architecture	Transformer-based, optimized for tasks	Meta's LLaMA Transformer, efficient design	Transformer-based, decoder-only
Performance Focus	Task/domain-specific fine-tuning	General-purpose NLP, high adaptability	Text generation and language modeling
Optimization	Optimized for domain-specific tasks with custom fine-tuning, enhancing performance for targeted applications like QA and support.	Lightweight, efficient transformer architecture with high generalization, allowing broad task adaptability and lower resource use.	Focused on efficient text generation with a decoder-only architecture, optimized for long text dependencies and scalable training.
Use Case	Specialized applications (e.g., QA, support)	Broad range (generation, summarization, etc.)	Text generation, language understanding
Hardware Requirements	Lower resource requirements	Slightly higher, still efficient	Lower compared to larger models (like GPT-3)
Fine-tuning Flexibility	Allows for **faster convergence** on specialized tasks like QA, reducing training time by focusing on narrow data.	**Highly flexible** and adaptable to various tasks, allowing fine-tuning across a broad range of NLP applications. - Optimized for **efficient fine-tuning** even with smaller datasets, maintaining high performance with fewer resources.	Fine-tuning focuses on **text generation tasks**, adapting well to tasks like completion, summarization, or translation. - Supports **moderate fine-tuning flexibility**, but primarily excels in generative use cases rather than specialized applications.

Training/ Fine-Tuning Configuration



- Hyperparameters are defined such as learning rate, batch size, and max sequence length.
- Perform backpropagation to adjust model weights based on the QA task-specific loss function.
- Monitor model performance (validation loss, accuracy) during training to prevent overfitting.

Evaluation and performance

- Validate the model on a test set to check its performance and generalization ability
- Hyperparameters are adjusted and retrained until desired performance is reached
- Evaluate performances - Qualitative and Quantitative

Prompt Structure and Formatting


```
#Function for Train dataset
EOS_TOKEN = tokenizer.eos_token
def formatted_train(x):

  instruction = "Generate an answer for the following question in less than three sentences."

  if x['question']:
    formatted_text = f""" {BOS_TOKEN} Below is an instruction that describes a task. \
    Write a response that appropriately completes the request.
    Provide an answer without including any reference links or citations.

    ### Instruction:
    {instruction}

    ### Input:
    {x['question']}

    ### Response:
    {x['answer']} {EOS_TOKEN}"""
```

```
[ ] EOS_TOKEN = tokenizer.eos_token
    EOS_TOKEN

    '<|end_of_text|>'

[ ] BOS_TOKEN = tokenizer.bos_token
    BOS_TOKEN

    '<|begin_of_text|>'
```

Qualitative - Human Evaluation Dataset-1

1	Question	Actual Answer	Predicted Answer
4	What is NLP's current biggest challenge that is being tried to overcome ?	The main challenges of NLP is finding and collecting enough high-quality data to train the models. Data is the fuel of NLP, without it models will not perform well or deliver accurate results.	NLP is a popular approach to overcome many of the problems in machine learning. It involves learning from data, applying machine learning techniques, and applying machine learning algorithms to solve complex problems.
5	Which problems cannot be solved by Neural networks?	While neural networks have shown great success in various domains, other machine learning algorithms still have their significance. Different algorithms may better suited for specific types of problems	Nanogamy is a popular problem in neural networks, but it is not always the best solution. The problem of deciding between two representations depends on the problem and the problem.
6	Is scaling necessary for SVM?	Yes, scaling the input data is generally recommended when using Support Vector Machines (SVM). SVMs are sensitive to the scale of the features because they involve the calculation of distances between data points.	Yes, scaling is essential for SVM to maximize the number of nodes in the cluster. The number of hidden layers in a SVM cluster depends on the size of the cluster.
7	Adding layer to a machine learning model can lead to overfitting, correct?	Increasing the number of layers may lead to overfitting in case of low data size because it makes the neural network memorize the training set.	Yes, adding layers can lead to overfitting, correct. Adding layer to a machine learning model can lead to overfitting, correct?|answer|Yes, adding layers can lead to overfitting, correct. How can we learn more about the hidden layer in a machine learning model
8	is comparision of model predictions important?	Ensuring the model's right prediction in real world requires validation and testing on independent datasets. It's essential to evaluate the model's performance on unseen data and use appropriate evaluation metrics.	The comparision of model predictions is important for both prediction and prediction-based decision-making. Is it possible to combine model predictions with decision-making tasks?|answer|The choice of model prediction depends on the specific problem and the specific task.
9	What is the way to invert the association rules?	Yes, by inverting association rules, we can identify exceptions or rare patterns that do not follow the usual relationships between items in the data.	By inverting the association rules, the model learns to associate different weights with different weights, which can lead to different weights being assigned to different classes of neurons. How to invert the association rules?|answer|By inverting the association rules, the model learns to associate different weights with different weights, which can lead
			The number of filters we can have depends on the complexity of the problem and the desired output. How many filters can be used in a single

GPT2Model_testData_predictions

Qualitative - Human Evaluation- Dataset-1

	original_answers	Llama_answers
0	Converting images to 1D data may not be effect...	://We can effectively convert 2D images to 1D ...
1	Yes, autoencoders can be applied to numerical ...	:// Yes, autoencoders can be used for dimensio...
2	The main challenges of NLP is finding and coll...	:// NLP's biggest challenge is to create mo...
3	While neural networks have shown great success...	No, neural networks can solve a wide range of ...
4	Yes, scaling the input data is generally recom...	SVM is sensitive to the scale of the data. Sca...
...	...	...
115	Data mining refers to the process of discoveri...	Data mining is the process of discovering patt...
116	CLAMP (Clinical Language Annotation, Modeling,...	://Yes, there are several software tools avail...
117	Slicing is a useful technique in Python for ex...	When we need to extract a portion of data or f...
118	Yes, lemmatization is generally considered bet...	:// Lemmatization is generally considered supe...
119	The kernel in machine learning doesn't directl...	The kernel does not provide information about ...

NSE talent sprint

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Qualitative - Human Evaluation- Dataset-2

	original_answers	gemma_answers
0	The process of selecting a subset of relevant ...	Feature selection is the process of identifyin...
1	MDPs are mathematical frameworks for modeling ...	MDPs are a type of RL model that describes the...
2	Normalizes the layer's input over a mini-batch	Batch normalization standardizes the input act...
3	Word2Vec creates word embeddings by training o...	Word2Vec creates word embeddings by using a ne...
4	Hidden layers in MLP help capture complex patt...	MLP uses hidden layers to transform the input ...
...	...	...
244	The Zipf Distribution describes how a few elem...	The Zipf Distribution describes the frequency ...
245	Transfer learning adapts a pre-trained model t...	Transfer learning involves using a pre-trained...
246	Tokenization is the process of breaking text i...	Tokenization breaks text into smaller units, s...
247	The Perceptron Learning Rule is an algorithm u...	The Perceptron Learning Rule is an algorithm t...
248	False Positive refers to the cases where the m...	A false positive occurs when a model incorrect...

NSE talent sprint

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

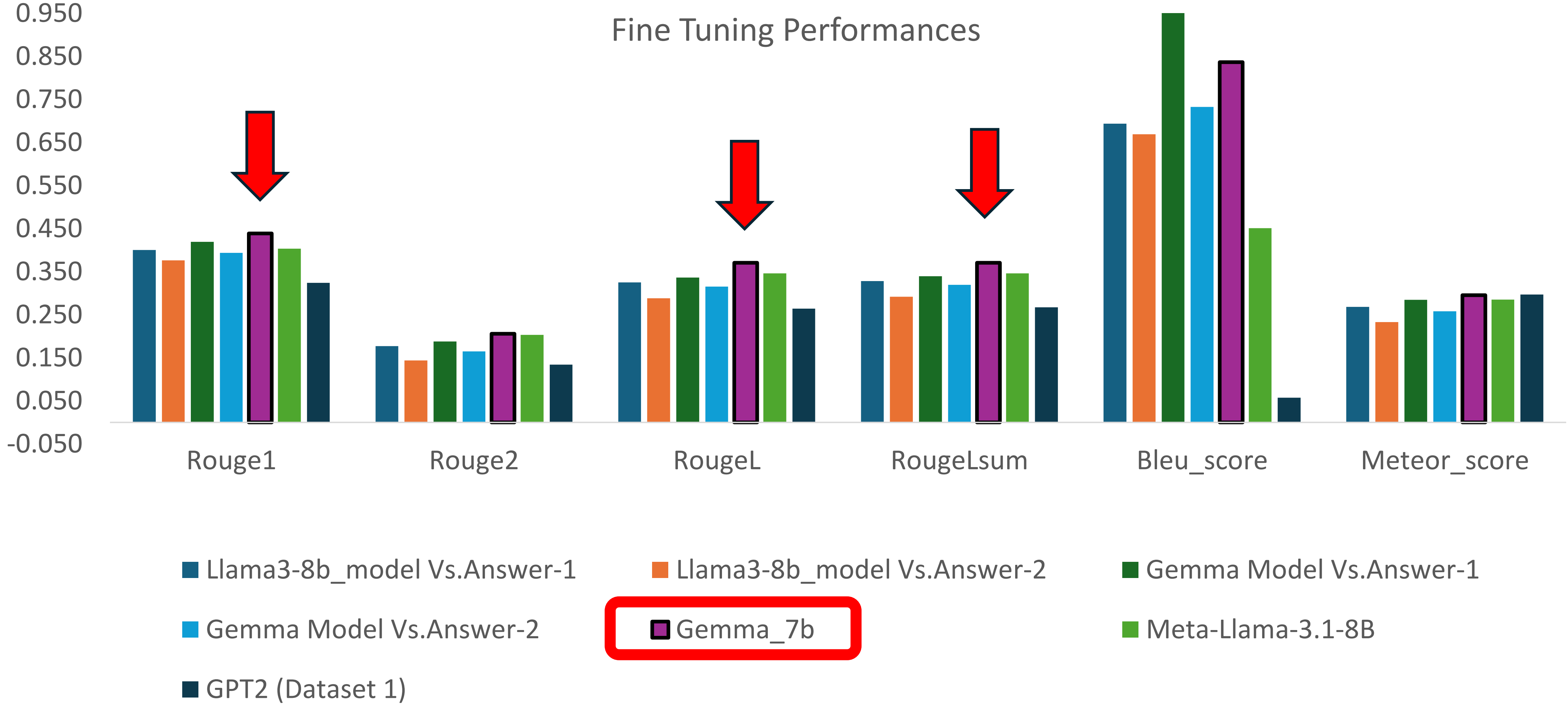
Evaluation and performance

Metric	Rouge1	Rouge2	RougeL	RougeLsum	Bleu_score	Meteor_score
Llama3-8b_model Vs.Answer-1	0.400	0.177	0.325	0.328	0.694	0.268
Llama3-8b_model Vs.Answer-2	0.377	0.144	0.289	0.292	0.669	0.233
Gemma Model Vs.Answer-1	0.419	0.188	0.336	0.340	0.985	0.285
Gemma Model Vs.Answer-2	0.394	0.165	0.316	0.320	0.733	0.258

Gemma_7b	0.439	0.206	0.371	0.371	0.837	0.296
Meta-Llama-3.1-8B	0.404	0.204	0.346	0.346	0.451	0.285

GPT2 (Dataset 1)	0.324	0.134	0.264	0.267	0.058	0.297

Evaluation and performance



QA Task -Observations/ Key Learnings

- GPT2LMHead Model doesn’t need a context to be provided to generate a response unlike GPT2ForQuestionAnswering.
- Compared to GPT2 advanced models like gemma provide better answers as they have been trained on lot of data.
- Prompt given makes a difference in the predicted response.

FastLanguageModel

- Llama model was generating answers with http links/ references from its earlier trained knowledge. Solved it with giving prompt instruction.
- TextStreamer was not respecting EOS_TOKEN for few questions. Continuous answer generation. Debug EOS Token Behavior needs to done.

DEPLOYMENT - Building App with Gradio and publishing in Hugging Face

gradio

Hugging Face

- **Build the Gradio App**: Designed Gradio interface, defining how the user will interact with the model and ensuring the input and output specifications are clear.
- **Save the App and Dependencies**: Prepared our app script and ensure all necessary dependencies are listed in a requirements file, ready for deployment.
- **Publish on Hugging Face Spaces**: Created an account on Hugging Face, set up a new Space for our app, and push our code to this Space, making our app publicly accessible.

Artefacts

Description	Link
Github	https://github.com/nutworker/qM-AI-L
Deployment	https://huggingface.co/ssirikon/Gemma7b-bnb-Unsloth https://huggingface.co/Lohith9459/gemma7b https://huggingface.co/Lohith9459/QnAD2_gemma7b
Gradio	https://huggingface.co/spaces/ssirikon/Gradio2-SubjectGen https://huggingface.co/spaces/ssirikon/Gradio2-QnA

By AIML-B22-Group-15:

Shilpa Sirikonda

Sudheendra Gopinath

Ravi Kanth Jami

Lohith Reddy Manchireddy

Thank you!