

# **PGCP - Capstone Project Report**

on

**Project Title: AI-based Generative QA System.**

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

**By:**

**AIML-B22-Group-15:**

Sirikonda Shilpa

Ravi Kanth Jami

Sudheendra Gopinath

Lohith Reddy Manchireddy

September 28, 2024  
Hyderabad

# INDEX

<a href="#">Introduction</a>	4
<a href="#">Email Subject Generation</a>	4
<a href="#">Dataset</a>	4
<a href="#">Data Loading and Pre-Processing</a>	4
<a href="#">Methodology</a>	5
<a href="#">Zero Shot Inferencing</a>	5
<a href="#">Finetuning Models</a>	6
<a href="#">Model Prompts – Illustrations</a>	8
<a href="#">Human Evaluation of Fine-tuned Models</a>	8
<a href="#">Quantitative Evaluation of Fine-tuned Models</a>	9
<a href="#">Granularity</a>	10
<a href="#">Context</a>	10
<a href="#">Summarization</a>	10
<a href="#">Evaluation against given subject lines</a>	11
<a href="#">Evaluation of Fine-tuning the pre-trained model</a>	11
<a href="#">Tuning the hyperparameters</a>	13
<a href="#">Observations &amp; Key Learnings</a>	14
<a href="#">Deployment</a>	15
<a href="#">Artefacts</a>	<b>Error! Bookmark not defined.</b>
<a href="#">Question Answering</a>	16
<a href="#">Model Approaches</a>	16
<a href="#">Dataset and Data Preparation</a>	16
<a href="#">Models' Selection</a>	17
<a href="#">Model Loading</a>	18
<a href="#">Environment Setup</a>	19
<a href="#">Training</a>	19
<a href="#">Prompt Structure and Formatting</a>	20
<a href="#">Unsloth &amp; QLora</a>	21
<a href="#">Qualitative / Human Evaluation</a>	22
<a href="#">Dataset-1 – GPT2</a>	22
<a href="#">Dataset 1 - Llama</a>	23
<a href="#">Dataset 1 - Gemma</a>	23
<a href="#">Dataset-2 - Gemma</a>	24
<a href="#">Issues noticed with Models' behaviour</a>	24

<a href="#">Repeated words in response</a>	25
<a href="#">Number sequence in response</a>	25
<a href="#">Zero Rouge score</a>	26
<a href="#">Code in response</a>	26
<a href="#">Fine Tuning Models</a>	26
<a href="#">Final Prompt Structure</a>	28
<a href="#">ROUGE Metrics</a>	28
<a href="#">Fine Tuning Performance</a>	29
<a href="#">Observations &amp; Key Learnings</a>	29
<a href="#">Save and Deploy</a>	29
<a href="#">Artefacts</a>	<b>Error! Bookmark not defined.</b>

## Introduction

GenAI QA System is a project to evaluate and compare the performance of various pre-trained generative AI models in NLP on two distinct tasks:

- given an email body, generate a succinct subject line for it
- answer technical questions on AI / ML.

## Email Subject Generation

GenAI QA System identifies the most salient words, phrases and sentences from the given email body and abstracts the message contained in that set into a very short, impactful subject line.

## Dataset

The pre-trained models are fine-tuned using the "[Annotated Enron Subject Line Corpus](#)" dataset.

- The dataset consists of a subset of cleaned, filtered and deduplicated emails from the Enron Email Corpus which consists of employee email inboxes from the Enron Corporation.

## Data Loading and Pre-Processing

- LangChain\_community.Directory Loaders are used to load the email files and then converted to Pandas DataFrame.(LangChain document\_loader was found to be organized, scalable, 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
- Some dataset statistics:
  - Sizes of train / dev / test splits: 14,436 / 1,960 / 1,906
  - An email contains an average of 75 words

- A subject contains an average of 4 words
- A subset of train dataset is created for finetuning language models, although full train data set is also used a couple of times.

## 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

## Zero Shot Inferencing

- Several models were loaded directly from hugging face and random records were inferenced to see how the models were behaving.
- [facebook/bart-base](#), [FLAN-T5](#) [Gemma-7b](#), [unsloth/mistral-7b](#), Phi-3 models were tried.
- By testing with various models with the Zero Shot Inferencing, we could see that the model struggles to extract the same subject line compared to the human baseline subject, but it does pull out some important information from the email which indicates the models can be fine-tuned to the task at hand.

Below is an example with Flan-T5 Base model.

```
-----  
INPUT PROMPT:  
  
Generate the subject line for the following email.  
  
Email:  
edmund i have reviewed the long descriptions i was given for the credit derivatives i cant tell if they are current :  
  
Subject:  
  
-----  
BASELINE ORIGINAL HUMAN ANNOTED SUBJECT LINE :  
EOL Credit Derivatives - Long Descriptions  
  
-----  
FLAN-T5-SMALL MODEL GENERATION - ZERO SHOT SUBJECT LINE:  
Credit Derivatives
```

## Finetuning Models

- Utilize the built-in Hugging Face/ SFTTrainer Trainer class. Pass the preprocessed dataset with reference to the original pretrained model. Several training parameters are tweaked and explored experimentally.
- Training a fully fine-tuned version of the basic/ small model takes a few hours on a GPU. To save time, several checkpoints were created, and the fully fine-tuned model were then initialized to use in the rest of experiments.

Some snapshots of the training process with the given data after pre-processing are below.

```
Map (num_proc=2):  0%|          | 0/11548 [00:00<?, ? examples/s]
max_steps is given, it will override any value given in num_train_epochs

trainer_stats = trainer.train()

==((====))==  Unsloth - 2x faster free finetuning | Num GPUs = 1
  \  / |      Num examples = 11,548 | Num Epochs = 1
0^0/ \_/ \   Batch size per device = 1 | Gradient Accumulation steps = 2
 \  / |      Total batch size = 2 | Total steps = 20
"-__-" |      Number of trainable parameters = 50,003,968
[20/20 00:54, Epoch 0/1]
```

Step	Training Loss
1	2.697600
2	2.816800
3	2.920200

```
# Train the model
facebook_bart_base_model_trainer.train()

Map:  0%|          | 0/2310 [00:00<?, ? examples/s]
Map:  0%|          | 0/462 [00:00<?, ? examples/s]
[867/867 10:58, Epoch 3/3]
```

Step	Training Loss	Validation Loss
200	7.010000	0.885521
400	0.353600	0.183391
600	0.220300	0.132789
800	0.154500	0.107520

Gopinath, Sudheendra (sgopinath@amd.com) is signed in

```
trainer.train()
```



[1734/1734 37:30, Epoch 3/3]

Epoch	Training Loss	Validation Loss
-------	---------------	-----------------

1	5.714500	0.152499
---	----------	----------

2	0.185400	0.139422
---	----------	----------

3	0.171000	0.137965
---	----------	----------

```
# Train the model
trainer.train()
```



[5775/8661 1:11:19 < 35:39, 1.35 it/s, Epoch 2/3]

Epoch	Training Loss	Validation Loss
-------	---------------	-----------------

1	1.644100	1.244680
---	----------	----------



[ 24/722 00:05 < 02:35, 4.50 it/s]



[8661/8661 1:51:40, Epoch 3/3]

Epoch	Training Loss	Validation Loss
-------	---------------	-----------------

1	1.644100	1.244680
---	----------	----------

2	1.912500	1.232297
---	----------	----------

3	1.766400	1.231620
---	----------	----------

```
TrainOutput(global_step=8661, training_loss=2.3930341022254935, metrics={'train_runtime':
'train_loss': 2.3930341022254935, 'epoch': 3.0})
```

## Model Prompts – Illustrations

Model	Prompt
<u>FLAN-T5</u>	<p>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 Subject as follows:</p> <p>Training prompt (email):</p> <p>prompt = f"" Generate a subject line for the following email.</p> <p>Email: {email}</p> <p>Subject:</p> <p>""</p>
<u>google-gemma-with-unsloth</u>	<div> <p>instruction = "Generate a subject line for the following email."</p> <p>if x['body']:</p> <p>formatted_text = f"" Below is an instruction that describes a task. \</p> <p>Write a response that appropriately completes the request.</p> <p>### Instruction:</p> <p>{instruction}</p> <p>### Input:</p> <p>{x['body']}</p> <p>### Response:</p> <p>{x['subject']}""</p> </div> <div> <p>Instruction = "Generate a subject line for the following email."</p> <p>If x['body']:</p> <p>formatted_text = f"" Below is an instruction that describes a task. \</p> <p>Write a response that appropriately completes the request.</p> <p>### Instruction:</p> <p>{instruction}</p> <p>### Input:</p> <p>{x['body']}</p> <p>### Response:</p> <p>""</p> </div>

## Human Evaluation of Fine-tuned Models

The performance of the fine-tuned models was qualitatively by comparing their ability to generate a reasonable subject line against its original subject, to asses if they are behaving the way they are supposed to, and are able to understand the input. This approach confirmed that the fine-tuned models behave as expected.



**\*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

**\*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

## Quantitative Evaluation of Fine-tuned Models

The ROUGE metric helps quantify the validity of subject lines produced by models. It compares subjects to an "annotated baseline" subject which is usually created by a human. While not perfect, it does indicate the overall increase in subject line generation effectiveness that we have accomplished by fine-tuning. The quantification is done on several dimensions such as below.

## Granularity

- ROUGE-1 focuses on individual words
- ROUGE-2 on word pairs
- ROUGE-L on the longest sequence of words

## Context

ROUGE-2 captures context better than ROUGE-1 due to its consideration of word pairs, while ROUGE-L and ROUGE-Lsum capture the overall sentence structure.

## Summarization

- ROUGE-Lsum is specifically designed for summarization, making it more relevant for evaluating the quality of summaries compared to ROUGE-L, which can be applied more generally.
- Bleu measures the precision: how much the words (and/or n-grams) in the machine generated summaries appeared in the human reference summaries.
- Rouge measures the recall: how much the words (and/or n-grams) in the human reference summaries appeared in the machine generated summaries.
- Rouge scores were evaluated for individual records and also as averages across all records.

Below table is evaluation for individual records.

```
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}
```

Below are the average-wise scores with Facebook/Bart model.

```

instruct_model_results = rouge.compute(
    predictions=trained_facebook_bart_subjects,
    references=human_annotated_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': 0.01409090909090909, 'rougeL': 0.06187822558905744, 'rougeLsum': 0.06115348980526484}
INSTRUCT MODEL:
{'rouge1': 0.365, 'rouge2': 0.13571428571428573, 'rougeL': 0.365, 'rougeLsum': 0.3716666666666667}

```

## Evaluation against given subject lines

The ROUGE scores were calculated w.r.t the original subjects and the three given human annotations.

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

## Evaluation of Fine-tuning the pre-trained model

The absolute percentage improvement of finetuned model over the pretrained model was also calculated.

🔗 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%

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

improvement = (np.array(list(ann2_finnetuned_model_results.values())) - np.array(list(ann2_ordin
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%

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

## Tuning the hyperparameters

Below are the key arguments used for training Gemma-7b.

- **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".

## 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

- **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: Question Answering

The QnA task is to generate free-form responses that require not only finding relevant information from its training knowledge but also synthesizing this information into multiple accurate answer sentences.

### Model Approaches

Both **extractive** and **sequence-to-sequence** approaches were explored for the given problem statement and fine-tuning. After careful evaluation, the 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.

### Dataset and Data Preparation

- Input-Output Pairs: For Seq2Seq fine-tuning the model is provided with structured input-output pairs, where:
  - Input: The question without a context
  - Output: The target answer.
- A total of 462 question-answer pairs were collaboratively prepared from the AIML course. The dataset was collated in the prescribed format in the CSV file.
- A consolidated train/dev/test set was provided for further fine-tuning with the GPT variant model.



- 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' Selection

GPT-2 medium, Gemma 7B and Llama 3 8B were used for finetuning. Below is a high level comparison of the models.

Feature	Gemma 7B	LLaMA 3 8B	GPT-2
<b>Model Size</b>	7 billion parameters	8 billion parameters	1.5 billion parameters
<b>Training Data</b>	Task-specific, proprietary data	General-purpose, large public datasets	Large-scale, web-based text data
<b>Architecture</b>	Transformer-based, optimized for tasks	Meta's LLaMA Transformer, efficient design	Transformer-based, decoder-only
<b>Performance Focus</b>	Task/domain-specific fine-tuning	General-purpose NLP, high adaptability	Text generation and language modeling
<b>Optimization</b>	Optimized for domain-specific tasks with custom fine-	Lightweight, efficient transformer architecture with	Focused on efficient text generation with a decoder-only architecture, optimized

	tuning, enhancing performance for targeted applications like QA and support.	high generalization, allowing broad task adaptability and lower resource use.	for long text dependencies and scalable training.
<b>Use Case</b>	Specialized applications (e.g., QA, support)	Broad range (generation, summarization, etc.)	Text generation, language understanding
<b>Hardware Requirements</b>	Lower resource requirements	Slightly higher, still efficient	Lower compared to larger models (like GPT-3)
<b>Fine-tuning Flexibility</b>	Allows for <b>faster convergence</b> on specialized tasks like QA, reducing training time by focusing on narrow data.	<b>Highly flexible</b> and adaptable to various tasks, allowing fine-tuning across a broad range of NLP applications. - Optimized for <b>efficient fine-tuning</b> even with smaller datasets, maintaining high performance with fewer resources.	Fine-tuning focuses on <b>text generation tasks</b> , adapting well to tasks like completion, summarization, or translation. - Supports <b>moderate fine-tuning flexibility</b> , but primarily excels in generative use cases rather than specialized applications.

## Model Loading

- Load the pretrained model (Gemma 7B or LLaMA 8B) from Huggingface using FastLanguageModel/ SFTTrainer class

- Create a PEFT model with the given parameters and load adapters - LoRA (Low-Rank Adaptation) for parameter-efficient fine-tuning with following parameters
  - `r=16`, # LoRa Rank
  - `target_modules=["q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj",]`,
  - `lora_alpha=16`,
  - `lora_dropout=0`,
  - `bias="none"`,
  - `use_gradient_checkpointing=True`

## Environment Setup

- Necessary libraries (e.g., PyTorch, Hugging Face Transformers) were set up
- Necessary GPU/TPU resources were made available for handling large models efficiently.

## Training

- Perform backpropagation to adjust model weights based on the task-specific loss function.
- Monitor model performance (validation loss, accuracy) during training to prevent overfitting.
- TrainingArguments used as below
  - `per_device_train_batch_size = 1`,
  - `gradient_accumulation_steps = 2`,
  - `warmup_steps = 5`,
  - `max_steps = 30`,
  - `learning_rate = 2e-4`,
  - `fp16 = not torch.cuda.is_bf16_supported()`,
  - `bf16 = torch.cuda.is_bf16_supported()`,
  - `logging_steps = 1`,

- optim = "paged\_adamw\_8bit",
- weight\_decay = 0.01,
- lr\_scheduler\_type = "linear",
- seed = 3407,
- output\_dir = "outputs",

## Prompt Structure and Formatting

- Added Clear Instructions: Specified format, tone, or length to guide the model's response.
- Provided Context: Included relevant details to anchor the model's answer closer to the reference.
- Used Examples: Provided sample responses to show the desired structure and style.

```
alpaca_prompt = """Below is an instruction that describes a task.
Write a response that appropriately completes the request.

### Instruction:
{}

### Input:
{}

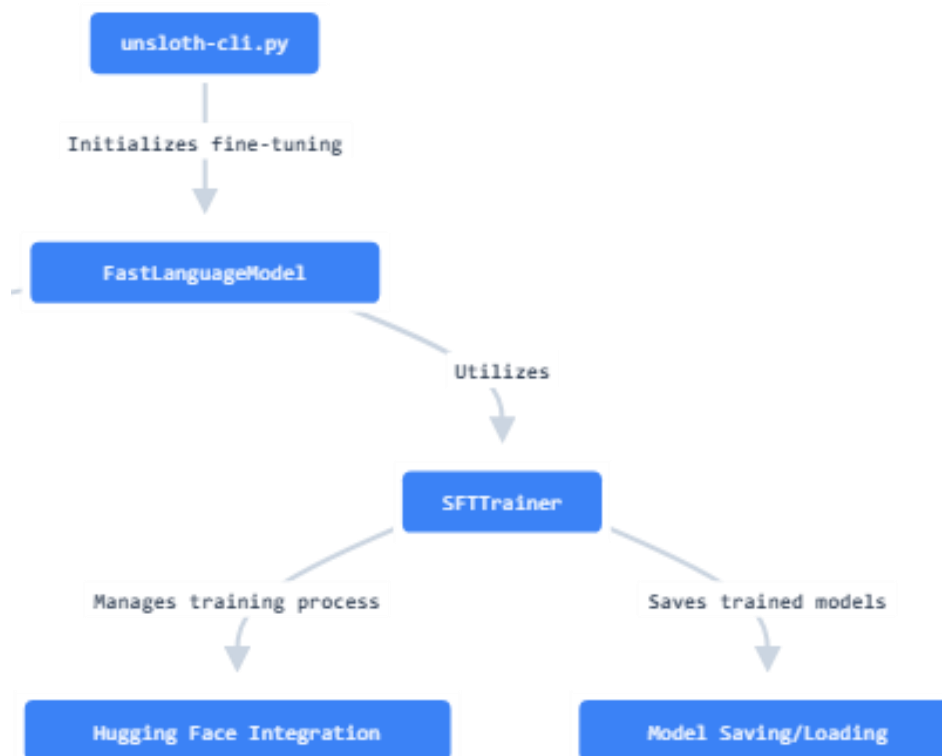
### Response:
{}"""

EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN
def formatted_train(x):
    instructions = """Generate an answer for the following question in less than three sentences.
    Provide an answer without including any reference links or citations.
    Do not include any special characters."""

    inputs      = x['question']
    outputs     = x['answer']
    # texts = []
    # for instruction, input, output in zip(instructions, inputs, outputs):
    #     # Must add EOS_TOKEN, otherwise your generation will go on forever!
    text = alpaca_prompt.format(instructions, inputs, outputs) + EOS_TOKEN
    # texts.append(text)
    return text
```

## Unsloth & QLora

There are many benefits in fine tuning the trained model with Unsloth and QLora.



- **Memory Efficiency:** Leveraging QLoRA and weight reduction techniques, FastLanguageModel minimizes memory consumption, enabling efficient fine-tuning on limited hardware.
- **Enhanced Speed:** Unsloth's Flash Attention via xformers, along with the use of causal masks, significantly speeds up training, allowing faster convergence without sacrificing performance.
- **Precision and Resource Optimization:** The Cross Entropy loss optimization in Unsloth reduces memory usage, ensuring high accuracy while maintaining computational efficiency.
- **Scalability:** The combination of bfloat16 and adaptive learning rates ensures that fine-tuning scales seamlessly, even on large datasets, with minimal resource requirements.

- **Cutting-edge Attention Mechanisms:** Unsloth integrates innovative attention techniques to further optimize transformer models, leading to improved model performance in a shorter time.

## Qualitative / Human Evaluation

Manually validate the model on a test set to check how its predictions compare with the actual answers.

## Dataset-1 – GPT2

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

## Dataset 1 - Llama

	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 ...

## Dataset 1 - Gemma

	answer-2	gemma_answers
0	To effectively convert 2D images to 1D, use te...	Reshaping the 2D image into a 1D vector is a c...
1	Yes, autoencoders can be used for dimensionali...	Yes, an autoencoder can be used for dimensiona...
2	NLP models struggle with tasks that require re...	The biggest challenge in NLP is dealing with t...
3	Neural networks are powerful, but they may str...	Neural networks can solve a wide range of prob...
4	Scaling the input data is advisable when utili...	Yes, scaling is necessary for SVM to ensure th...
...	...	...
115	Machine learning is a subset of data mining th...	Data mining is a process of discovering patter...
116	CLAMP is a comprehensive clinical Natural Lang...	Yes, there are several tools available for cli...
117	Slicing can be useful for working with large d...	Slicing is used to extract a specific part of ...
118	Yes,Unlike stemming, which simply trims words ...	In general, lemmatization is considered to be ...
119	No. The kernel is a function used in machine l...	The kernel in a neural network provides inform...



## Dataset-2 - Gemma

	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...

	original_answers	Llama_answers
0	Words with similar meanings are represented by...	1n2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 1...
1	Variance refers to the error introduced by the...	Variance is a measure of the spread of the dat...
2	A cross-validation score is the performance me...	:// A cross-validation score is a metric used ...
3	PEFT refers to techniques that allow fine-tuni...	php require_once("php/answer.php");?> Paramete...
4	Feature selection helps prevent overfitting by...	php echo htmlspecialchars(\$response);? Featur...
5	A fully connected layer connects each neuron t...	php echo " A fully connected layer in deep le...
6	GPT-3 is an advanced version of GPT-2, with 17...	:// GPT-3 is a more powerful version of GPT-2,...
7	TF-IDF is the product of TF and IDF.	:// TF-IDF is calculated by multiplying the t...
8	Feature engineering for unstructured data invo...	:// Feature engineering for unstructured da...
9	Eigenvalues and eigenvectors are scalar values...	2nbsp;Eigenvalues are the values that are asso...
10	Supervised learning involves data with labels,...	:// Supervised learning requires labeled data...
11	A hyperplane is a decision boundary that separ...	:// A hyperplane in the context of SVM is a de...

## Issues noticed with Models' behaviour

Some issues that were noticed with the behaviour of the models right after training with out dataset and before tweaking and fine-tuning the hyperparameters are listed below.



## Repeated words in response

The answer to a question sometimes contains a set of words repeated many times. The phrase that is repeated, is also mostly irrelevant to the question asked.

```
### Instruction:
Generate an answer for the following question in less than three sentences.

### Input:
what are the evaluation methods to determine, if the images have been sufficiently denoised?

### Response:
<|end_of_text|><|begin_of_text|>://https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/
https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggl
e.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://ww
w.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/http
s://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.co
m/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kag
gle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://w
ww.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/htt
ps://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.c
om/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kag
gle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://
www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/ht
tps://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.
com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle.com/https://www.kaggle
```

## Number sequence in response

Sometimes the answer to a question just contains numbers from 1 to 256, increasing monotonically.

```
### Instruction:
Generate an answer for the following question in less than three sentences.

### Input:
How do word embeddings capture semantic relationships between words?

### Response:
<|end_of_text|><|begin_of_text|>1n2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 6
4 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 10
0 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 1
55 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 18
2 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 2
37 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256
```

## Zero Rouge score

Sometimes the answer to a question is so off from the expected answer that the Rouge score is zero.

9	Eigenvalues and eigenvectors are scalar values...	The eigenvalues and eigenvectors of a matrix a...	0.478261	0.227273	0.434783	0.434783	1.199348	0.359569
10	Supervised learning involves data with labels,...	php session_start(); if (isset(\$_SESSION['sess...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
11	A hyperplane is a decision boundary that separ...	php print("A hyperplane is a dividing plane th...	0.326531	0.212766	0.326531	0.326531	1.506189	0.207825
12	The discount factor is a parameter that determ...	* the discount factor is the discount rate per...	0.434783	0.285714	0.434783	0.434783	3.386499	0.290625
13	Extract the text from the HTML page without an...	// To extract text from an HTML page using Be...	0.346154	0.120000	0.269231	0.307692	0.813035	0.379189
14	Regression involves predicting a real number w...	php print_r(\$response);?	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15	An n-gram where n=1, meaning a single word.	php include_once("/home/word2vec/word2vec.php");?	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

## Code in response

Sometimes the answer contains code.

```
### Instruction:
Generate an answer for the following question in less than three sentences.
Provide a answer without including any reference links or citations.

### Input:
What is the primary difference between supervised and unsupervised learning?

### Response:
<|end_of_text|><|begin_of_text|>php session_start(); if (isset($_SESSION['session'])) { $session = $_SESSION['session']; } else { $session = ''; }<|end_of_text|>
```

This is more likely when there is some word, like "HTML" in the prompt.

```
### Instruction:
Generate an answer for the following question in less than three sentences.
Provide an answer without including any reference links or citations.
Do not include any special characters

### Input:
How to extract text from an HTML page using BeautifulSoup?

### Response:
<|end_of_text|><|begin_of_text|> = BeautifulSoup(response, "html.parser")<|end_of_text|>
```

## Fine Tuning Models

- Couple of hyperparameters were selected and tweaked such as learning rate, optim, max sequence length, lr\_scheduler\_type etc.,
- 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.

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

The below tables depict how the various hyperparameters were tweaked to push the performance of the models higher.



Avg Scores	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Exp-6	Exp-7	
Rouge1	0.3096	0.3282	0.3332	0.3468	0.3766	0.4226	<b>0.4352</b>	13%
Rouge2	0.1216	0.1362	0.1405	0.1385	0.1439	0.182	<b>0.1972</b>	8%
RougeL	0.2341	0.273	0.2645	0.2863	0.2885	0.3506	<b>0.3643</b>	13%
RougeLsum	0.2368	0.2749	0.2683	0.2885	0.2918	0.3506	<b>0.3643</b>	13%
Bleu_score	0.4413	0.634	0.3627	1.2561	0.6694	0.4776	<b>0.5545</b>	11%
Meteor_score	0.2342	0.273	0.3046	0.2534	0.2329	0.3135	<b>0.3228</b>	9%



max_seq_length	512	1024	1024	1024	1024	1024	1024
max_new_tokens	50	100	100	100	100	100	200
optim	adamw_8bit				paged_adamw_8bit		
max_step	10	10	10	30	20	20	20
learning_rate	2e-4	2e-4	2e-4	3e-4	2e-4	2e-5	2e-6
temperature	default						0.6
top_p	default						0.9
lr_scheduler_type	Linear	Linear	Linear	Linear	cosine	Linear	Linear
r	16	16	16	32	16	16	16
prompt	Without eos_token_id	With eos_token_id	+ "without reference & citations prompt"	+ "Without Special Characters"			

## Final Prompt Structure

```
alpaca_prompt = """Below is an instruction that describes a task.
Write a response that appropriately completes the request.

### Instruction:
{}

### Input:
{}

### Response:
{}"""

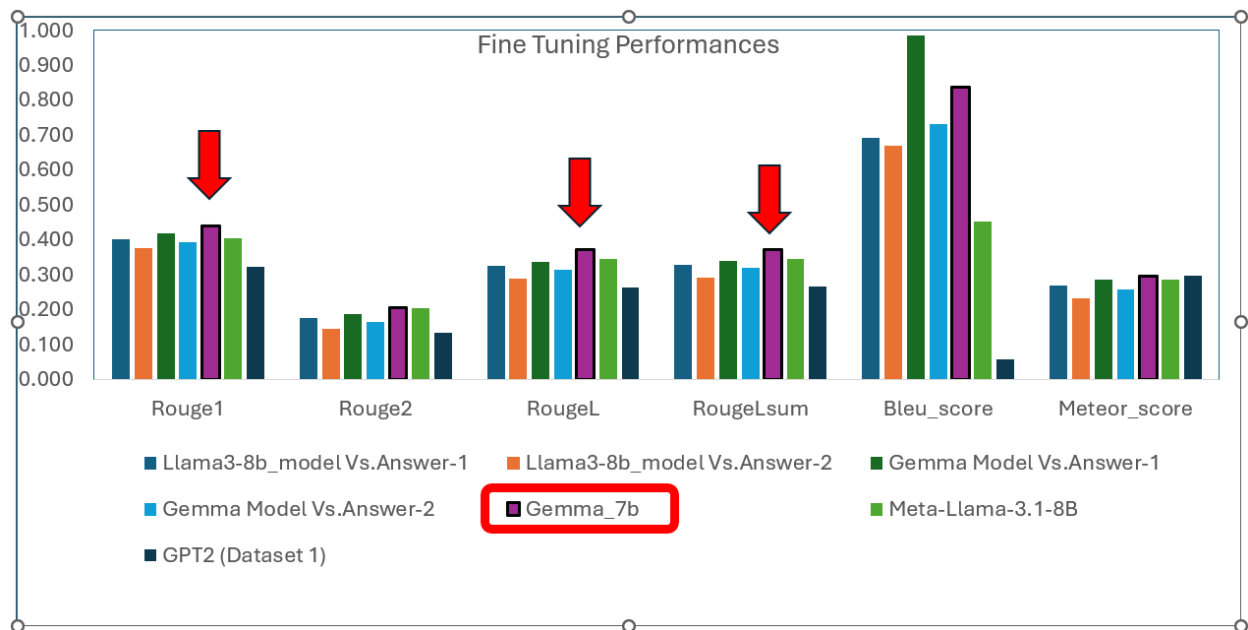
EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN
def formatted_train(x):
    instructions = """Generate an answer for the following question in less than three sentences.
    Provide an answer without including any reference links or citations.
    Do not include any special characters."""

    inputs      = x['question']
    outputs     = x['answer']
    # texts = []
    # for instruction, input, output in zip(instructions, inputs, outputs):
    #     # Must add EOS_TOKEN, otherwise your generation will go on forever!
    text = alpaca_prompt.format(instructions, inputs, outputs) + EOS_TOKEN
    # texts.append(text)
    return text
```

## ROUGE Metrics

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
<b>Gemma_7b</b>	<b>0.439</b>	<b>0.206</b>	<b>0.371</b>	<b>0.371</b>	<b>0.837</b>	<b>0.296</b>
Meta-Llama-3.1-8B	0.4352	0.1972	0.3643	0.3643	0.5545	0.3228
GPT2 (Dataset 1)	0.324	0.134	0.264	0.267	0.058	0.297

## Fine Tuning Performance



## 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, llama provide better answers as they have been trained on lot of data.
- Prompt given makes a difference in the predicted response.
- 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 in random before finetuning. Continuous answer generation was noticed. Debugged EOS Token initialized at right places.

## Save and Deploy

- 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	<a href="https://github.com/nutworker/qM-AI-L">https://github.com/nutworker/qM-AI-L</a>
Deployment	Task-1: Email Subject Line Generation <a href="https://huggingface.co/ssirikon/Gemma7b-bnb-Unsloth">https://huggingface.co/ssirikon/Gemma7b-bnb-Unsloth</a> <a href="https://huggingface.co/Lohith9459/gemma7b">https://huggingface.co/Lohith9459/gemma7b</a>  Task-2: Q&A Task <a href="https://huggingface.co/Lohith9459/QnAD2_gemma7b">https://huggingface.co/Lohith9459/QnAD2_gemma7b</a>
Gradio	<a href="https://huggingface.co/spaces/ssirikon/Gradio2-SubjectGen">https://huggingface.co/spaces/ssirikon/Gradio2-SubjectGen</a> <a href="https://huggingface.co/spaces/ssirikon/Gradio2-QnA">https://huggingface.co/spaces/ssirikon/Gradio2-QnA</a>

**\*\* \*\* \***

## Summaries of three Research Papers

### 1. "Attention Is All You Need" (2017) by Vaswani et al.

**Summary:** This paper introduced the Transformer architecture, which revolutionized natural language processing (NLP) and has since become the foundation for many state-of-the-art models (e.g., BERT, GPT, LLaMA). The key innovation is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to each other, regardless of their position. Unlike previous models like RNNs and LSTMs, the Transformer eliminates the need for sequential data processing, enabling faster training and better scalability for large datasets.

**Impact:** It shifted the paradigm from recurrent networks to attention-based mechanisms, fundamentally changing how we approach NLP and sequence modeling tasks.

---

### 2. "Generative Adversarial Nets" (2014) by Ian Goodfellow et al.

**Summary:** This paper introduced Generative Adversarial Networks (GANs), a groundbreaking model that pits two neural networks against each other: a generator that creates data, and a discriminator that evaluates the authenticity of the generated data. This adversarial training enables GANs to produce highly realistic synthetic data, such as images, videos, and music.

**Impact:** GANs have become foundational for generative tasks in AI, influencing areas like image synthesis, data augmentation, and unsupervised learning, spurring a wave of innovation in creative AI applications.

---

### 3 "Language Models are Few-Shot Learners" (2020) by Brown et al. (GPT-3)

**Summary:** This paper introduced GPT-3, a large-scale language model capable of generating coherent and contextually relevant text with minimal input. The paper highlighted the model's ability to perform a wide range of language tasks (translation, summarization, question-answering) through few-shot learning, without the need for extensive task-specific training.

**Impact:** GPT-3 demonstrated the power of massive, pre-trained models, pushing the boundaries of what generative AI can achieve in tasks like writing, coding, and creative content generation. It paved the way for models like GPT-4 and other advanced generative AI systems.