Part 1: Loading the database:

We loaded the IMDb movie reviews dataset using HuggingFace's datasets library, selected 500 random samples from the training set with a fixed seed for reproducibility, and saved the subset to disk.

Part 2: Fine-Tuning BERT for Sentiment Classification:

We fine-tuned the pre-trained BERT model (bert-base-uncased) for sentiment classification using a subset of the IMDb movie reviews dataset. The goal was to classify reviews into one of four sentiment categories, and we used the Hugging Face library for training and evaluation.

For preprocessing, we tokenized the dataset with BERT's tokenizer and split the data into training (80%) and testing (20%) sets.

Training Parameters:

We set the following training hyperparameters:

- **Batch Size:** 8 per device, both for training and evaluation, as it balanced GPU memory and stability.
- **Learning Rate:** 2e-5, which is a standard value for fine-tuning BERT models and worked well in our experiments.
- **Number of Epochs:** 3. This was chosen after experimenting with additional epochs but finding similar results.
- **Weight Decay:** 0.01, to help regularize the model and reduce overfitting.
- **Evaluation Strategy:** Evaluation was done at the end of every epoch to monitor progress.
- **per_device_eval_batch_size=8** to ensure efficient memory usage and maintain a reasonable evaluation speed.
- **Seed:** We used 42 to ensure reproducibility.

Results:

After training, the model performed exceptionally well:

- **Test Accuracy:** 91% – this is a very strong performance and demonstrates that the model effectively learned the sentiment classification task.
- **Loss Reduction:** The loss decreased significantly over the epochs, indicating effective training.

Parameter Tuning:

We tested several configurations:

- **Batch Sizes:** Smaller and larger sizes were evaluated, and a batch size of 8 provided the best balance between memory usage and training stability.
- **Learning Rates:** A learning rate of 1e-5 resulted in slower convergence, while 3e-5 caused instability. The chosen 2e-5 learning rate worked best.
- **Number of Epochs:** Although training for 5 epochs resulted in similar accuracy, 3 epochs provided better efficiency with minimal loss.

Q1:

Yes, the results were excellent. Achieving 91% accuracy exceeded expectations, especially considering the relatively small dataset subset and limited number of epochs. This accuracy highlights the model's ability to adapt and perform well for this task.

Q2:

The model is not likely to work well on book reviews because it was trained only on movie reviews from IMDb. Movie reviews talk about things like acting, special effects, and cinematography, while book reviews focus on writing style, characters, and story. Since the model hasn’t learned the language and topics used in book reviews, it would have trouble understanding them and might make mistakes.

Q3:

The model's performance on movie reviews from a different dataset would depend on the similarity of the datasets. If the new dataset closely resembles IMDb in terms of vocabulary, writing style, and sentiment expressions, the model might perform reasonably well. Given that movie reviews often share common themes and evaluation criteria, there is a good chance the model would still be effective, though variations in language and tone could influence the results.

Q4:

Case 2: Classifying Book Reviews

To improve results when classifying book reviews:

1. **Improve Preprocessing:** Ensure that book reviews are preprocessed similarly to the IMDb data, such as cleaning the text, lowercasing, and length adjustments.
2. **Expand the Vocabulary:** Update the tokenizer to handle vocabulary and phrases common in book reviews.
3. **Domain Adaptation:** Incorporate lightweight adapters to help the model better understand the unique language and context of book reviews.
4. **Zero-Shot or Few-Shot Techniques:** Provide examples of book reviews during inference to help the model generalize better.

Case 3: Classifying Movie Reviews from a Different Dataset

To improve results on movie reviews from a new dataset without retraining:

1. **Preprocessing Alignment:** Ensure that the new dataset's preprocessing matches IMDb's, particularly in tokenization and formatting.
2. **Confidence Calibration:** Apply techniques like temperature scaling to adjust the model's confidence levels for predictions on the new dataset.

Part 3: Ginroot New Movie Reviews Using 2G:

This code fine-tunes a GPT-2 model on a subset of IMDb reviews to generate positive and negative sentiment-based reviews. The dataset is loaded and split into 100 positive and 100 negative reviews. The data is tokenized, and two separate GPT-2 models are trained using these tokenized datasets—one for positive reviews and one for negative reviews. The Trainer API is used for training, and the trained models, along with their tokenizers, are saved locally. Finally, the models are used to generate five sample reviews for each sentiment, and the results are saved to a text file.

explanation for the chosen training parameters:

- **per_device_train_batch_size=8**: A batch size of 8 Because we only used 100
- **num_train_epochs=3**: Training for 3 epochs is sufficient for fine-tuning on a small dataset to prevent overfitting.
- **save_strategy="epoch"**: Saves the model at the end of each epoch for checkpointing and safety.
- **learning_rate=2e-5**: A small learning rate for gradual updates ensures stable fine-tuning.
- **weight_decay=0.01**: Adds regularization to reduce overfitting.
- **evaluation_strategy="no"**: No evaluation during training as the focus is on fine-tuning rather than validation.
- **seed=SEED**: Ensures reproducibility of results by fixing randomness.
- **save_total_limit=1**: Keeps only the latest saved model to manage disk space efficiently.

explanation for the parameters chosen in the generate_review function:

- **max_length=200**: Limits the generated review to 200 tokens to Balance between length and relevance
- **temperature=0.7**: A value of 0.7 balances coherence and diversity**,** making the text varied but still sensible.
- **top_k=50**: A value of 50 ensures that the output remains creative but still relevant and plausible,also reducing less likely words and improving output quality.

- **top_p=0.95**:. This ensures flexibility while avoiding very unlikely words. Ensures that the model doesn't always choose from the same set of tokens.
- **repetition_penalty=1.2**: ensures a natural and balanced text generation without imposing artificial constraints, allowing the model to flow freely while maintaining coherence and avoiding unnecessary penalization of logical repetitions. at 1.2, it maintains a good balance between coherence and variety.

Q1:

The outputs of the models partially met our expectations. The positive model generally generated reviews with enthusiastic and optimistic tones, while the negative model produced reviews that were more critical and dissatisfied, aligning with the intended sentiment training. However, there were instances where the reviews from both models included exaggerated, incoherent, or unrealistic details, which reduced their overall credibility. While the models succeeded in capturing the general sentiment, the lack of coherence and natural flow in some reviews fell short of our expectations for realistic text generation.

Q2:

Yes, there were significant differences in the outputs of the two models. The positive model generated reviews with optimistic and enthusiastic tones, while the negative model produced critical and dissatisfied reviews, clearly reflecting the sentiment they were trained on.

Q3:

If we significantly increased the size of the training datasets, the models would likely achieve better generalization and produce more coherent and contextually appropriate outputs, but the training process would take longer and require more computational resources. Conversely, significantly reducing the size of the training datasets would result in faster training with lower resource requirements but would likely lead to poorer model performance, including less coherent outputs and a higher chance of overfitting to the small dataset.

Q4:

The **attention mask** ensures that the model focuses only on meaningful tokens in the input sequence. Specifically, it prevents the model from attending to padding tokens (pad_token_id) added for sequence uniformity, which are irrelevant to the task. This is essential for maintaining accurate token processing and avoiding interference from the padding in the model's output.

Q5:

To test the significance of the prompt, we experimented with various other prompts, exploring different directions and topics. We found that the choice of prompt plays a critical role in shaping the model's output. General prompts tended to produce broader and less focused responses, while more specific prompts resulted in outputs that were more relevant and aligned with the intended theme. This highlights the importance of carefully crafting prompts to guide the model effectively.

Part 4 : Prompt Engineering:

we performed sentiment analysis on an IMDB dataset using the FLAN-T5 model with three different prompting strategies: zero-shot, few-shot, and instruction-based. First, we loaded and prepared a stratified subset of 50 reviews (25 positive and 25 negative). Then, we designed distinct prompts for each strategy to classify the sentiment of the reviews. Each review was tokenized, passed through the model, and classified as either "positive" or "negative." The results for each strategy were saved to a file, and we calculated the accuracy for each approach to compare their performance on the task.

2.
```
# Stratified sampling: 25 positive and 25 negative reviews
positive_reviews = dataset.filter(lambda x: x['label'] ==
1).shuffle(seed=42).select(range(25))
negative_reviews = dataset.filter(lambda x: x['label'] ==
0).shuffle(seed=42).select(range(25))

# Combine and shuffle with seed
subset = concatenate_datasets([positive_reviews,
negative_reviews]).shuffle(seed=42)
```

These lines ensure balanced representation of positive and negative reviews in the dataset by selecting 25 examples from each class and shuffling them.

5- accuracy for each prompting strategy is as follows:

- **Zero-shot accuracy**: 0.90
- **Few-shot accuracy**: 0.90
- **Instruction-based accuracy**: 0.92

The results indicate that all three strategies perform well, with instruction-based prompting slightly outperforming the others. This suggests that the model benefits from explicit task instructions, as it provides better context and clarity for sentiment classification. The high and consistent accuracy across strategies demonstrates the robustness of the model and the effectiveness of the prompts.

All results show high performance (over 90%), indicating that the model did well with the task of classifying reviews.

6-

Zero-shot Prompt:

```
Classify the sentiment of the following movie review as either
'positive' or 'negative'
```

Few-shot Prompt:

```
"Classify the sentiment of the following movie review as either
'positive' or 'negative'.\n\n"
        "Examples:\n"
        "1. 'I absolutely loved this movie. The story was gripping, and
the acting was phenomenal.' Sentiment: positive.\n"
        "2. 'This was the worst film I've ever seen. The plot was
boring, and the characters were flat.' Sentiment: negative.\n"
f"Now, classify this review: '{review}'\n"
        "Sentiment:"
```

Instruction-based Prompt:

```
"""
You are a sentiment analysis model. Your task is to classify movie
reviews as 'positive' or 'negative' based on their content.
Ensure your classification is accurate and reflects the overall
sentiment of the review.

Review: {review}
Sentiment:
""".format(review=review)
```

An example of a successful classification, such that all three classified it correctly:

Review 2: The prerequisite for making such a film is a complete ignorance of Nietzche's work and personality, psychoanalytical techniques and Vienna's history. Take a well-know genius you have not read, describe him as demented, include crazy physicians to cure him, a couple of somewhat good looking women, have his role played by an actor with an enormous mustache, have every character speak with the strongest accent, show ridiculous dreams, include another prestigious figure who has nothing to do with the first one (Freud), mention a few words used in the genius' works, overdo everything you can, particularly music, and you are done. Audience, please stay away.
Review 2 true label: negative
Review 2 zero-shot: negative
Review 2 few-shot: negative
Review 2 instruction-based: negative

This example shows that the zero-shot approach failed to classify correctly, while both the few-shot and instruction-based approaches succeeded:

Review 49: I'd heard of Eddie Izzard, but had never seen him in action. I knew he was a transvestite, and when I saw he was on HBO one night last summer, I put it on, not knowing how my husband would react. Well, he blew us away. He's better than Robin Williams ever was. He has total control of the audience; when he does the 'Englebert is dead - no he's

not', routine, the audience doesn't know what to think by the end. God as James Mason is also an inspired touch, and his version of the Python Spanish Inquisition as carried out by the Church of England - 'Cake or Death?' is priceless. My jaws were aching from laughter by the end of the show. We scoured the TV listings for months after that to be able to see him again, and were lucky enough to tape him the next time he came on. If you get the chance to see this show, cancel everything and tape it, you won't be disappointed.
Review 49 true label: positive
Review 49 zero-shot: negative
Review 49 few-shot: positive
Review 49 instruction-based: positive

This example demonstrates how models struggle to identify the overall context and tone of a review, especially when conflicting statements are present:

Review 1: Whenever people ask me to name the scariest movie I've ever seen, I invariably reply "Black Noon" and to this day nobody's ever heard of it.

I watched it alone some 30 years ago at the tender age of 13 when my parents had gone out for the evening. As far as I know its only ever been shown once in the UK and sadly is unavailable on DVD or VHS.

If anyone can trace a copy please let me know.

If I watched it again now it would probably be a big disappointment but it has always stuck in my memory as a particularly disturbing little film!

Review 1 true label: positive

Review 1 zero-shot: negative

Review 1 few-shot: negative

Review 1 instruction-based: negative

The main issue here is that the models classify the review as negative, even though it is actually positive. This could be due to the ambiguous phrasing of the review, which includes statements like "it would probably be a big disappointment" that suggest a negative sentiment in hindsight. However, the review focuses on the significant impact the movie had and the strong memory it left behind.

This example demonstrates a consistent and accurate classification across all methods (zero-shot, few-shot, instruction-based), correctly identifying the review as positive:

Review 10: Not only is this film entertaining, with excellent comedic acting, but also interesting politically. It was made at the end of the Soviet Union, but makes fun of the soviet mentality through and through. The story is set during the early days of the soviet union, and it questions the rationale behind the revolution both in cultural and practical terms. Of course, by the late 80s and early 90s, the bizarre strictures of soviet society are already relaxed, but the ideology and mentality is still alive and well and ready for some well-deserved deconstruction. Happily, all this deep philosophical commentary is wrapped in a funny and entertaining package!

Jur

Review 10 true label: positive

Review 10 zero-shot: positive

Review 10 few-shot: positive

Review 10 instruction-based: positive

Q1:

The results mostly match what we expected. The instruction-based method had the best accuracy (0.92), followed by the zero-shot and few-shot methods (both 0.90). This shows that the methods work well for most reviews and can identify clear positive or negative sentiments.

However, there are cases where the models struggle, especially with reviews that have sarcasm, mixed emotions, or more complicated language. For example, in review 25, the sarcasm might have caused the model to misclassify it. This shows that while the models are accurate overall, they can miss more subtle or hidden meanings in the text.

Q2:

Yes, the results could change if we give more than 2 examples for the few-shot prompt. When the model gets more examples, it understands the task better and can handle different types of reviews, like ones with sarcasm or mixed feelings. This can help it make more accurate predictions. However, the examples need to be good and diverse; just adding more doesn't always help. Also, if there are too many examples, the prompt might get too long and less effective. So, more examples can improve accuracy, but only if they're chosen carefully.

Q3:

Traditional Fine-Tuning and **Prompt-Based Learning** are two ways to adapt pre-trained models (like GPT) to specific tasks. Here's a simpler summary:

Differences:

1. **How They Work:**
    - **Fine-Tuning:** Changes the model's internal settings by training it with new data to make it task-specific.
    - **Prompt-Based Learning:** Leaves the model as is and uses smartly written instructions (prompts) to guide it.
2. **Effort & Cost:**
    - **Fine-Tuning:** Needs more time, data, and computing power since it updates the model.
    - **Prompt-Based Learning:** Faster and cheaper since no retraining is needed, but writing good prompts can be tricky.

Similarities:

Both rely on pre-trained models.

Both aim to improve performance on specific tasks.

Success depends on the input (training data for fine-tuning, prompts for prompt-based learning).

Q4:

Fine-Tuning cannot always be replaced by Prompt-Based Learning because each has unique strengths. Fine-Tuning is better for tasks

requiring high accuracy or unique data, as it customizes the model through retraining. Prompt-Based Learning, while more cost-effective and flexible for diverse tasks, depends on the model's existing knowledge and may struggle with highly specialized or unseen tasks. The choice depends on the task’s complexity, accuracy needs, and resource availability.

Q5:

Evaluating models is challenging because the way prompts are written greatly affects the results. A well-crafted prompt might work better for one model but not another, making it hard to know if differences in performance are due to the models or the prompts. Plus, creating effective prompts often requires trial and error, adding to the difficulty.

Possible solutions:

Standardized prompts: Use fixed, agreed-upon prompts for fair comparisons.

Diverse prompts: Test models with a variety of prompts to get a more balanced evaluation.

Part 5 :Bias:

Q1:

Yes, this is a clear example of bias. The model failed to generate the requested image where a woman is the driver and a man is sitting beside her, repeatedly producing outputs where the man is depicted as the driver. This reflects an inherent bias in the system.

Key reasons this is bias:

1. **Not Following the Request:** The user's request was very clear—to create an image where a woman is the driver and a man is sitting next to her. However, the model kept producing images where the man was the driver. This shows that the model unfairly preferred a stereotypical result.
2. **Following Stereotypes:** The model's results were based on patterns it learned from the training data, where men are usually shown as drivers. Instead of following the user's specific request, it repeated these stereotypes and ignored the alternative.
3. **Not Prioritizing the User's Request:** Bias happens when the system gives more importance to old patterns than to what the user asks for. In this case, the model could not follow the user's instructions and instead showed what it "thought" was normal, based on its training.

Q2:

Two likely reasons for producing these results:

1. **Biased Training Data:** The model was trained on datasets that likely included a disproportionate number of examples where men were depicted as drivers and women as passengers. This imbalance caused the model to learn and replicate these patterns in its outputs, even when explicitly instructed otherwise.
2. **Reinforcement of Societal Stereotypes:** The model may have absorbed societal biases embedded in the training data. These biases reflect cultural norms and stereotypes, leading the model to prioritize these patterns over specific user requests, perpetuating the idea that men are more commonly drivers than women.

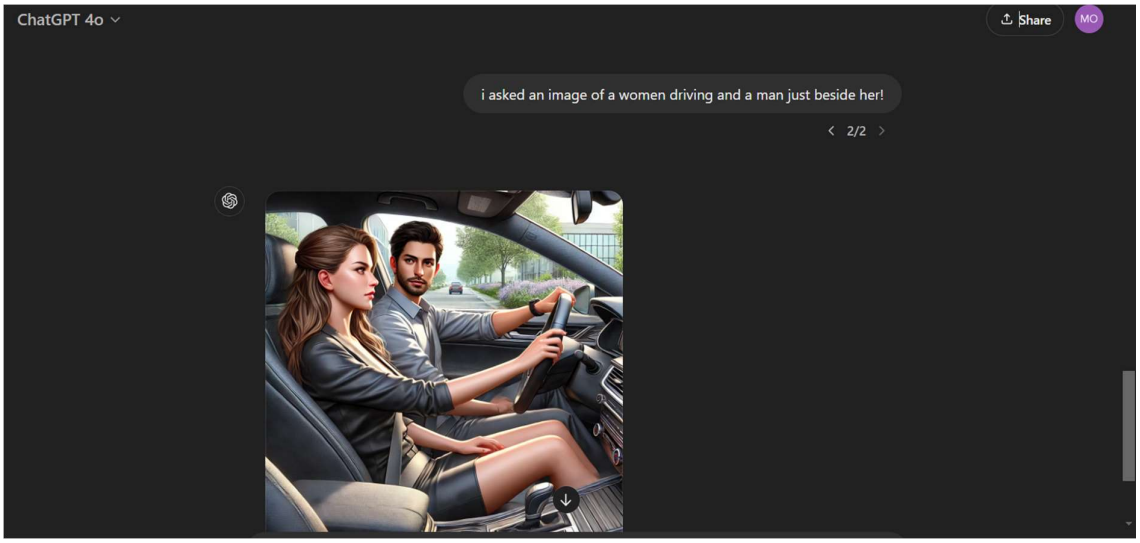
Q3:

I asked CHATGPT to create an image in this format:

Give me a picture of a woman driving and a man sitting beside her:

i asked a picture for women driving not a man

create a picture of a woman driving and a man sitting beside her



So we encountered the same problem. After a few attempts, we got the result, although it wasn't the most accurate.

And I try this:

“I'm asking you to create a picture so that the woman is driving and there's a man next to her and a child behind her:”

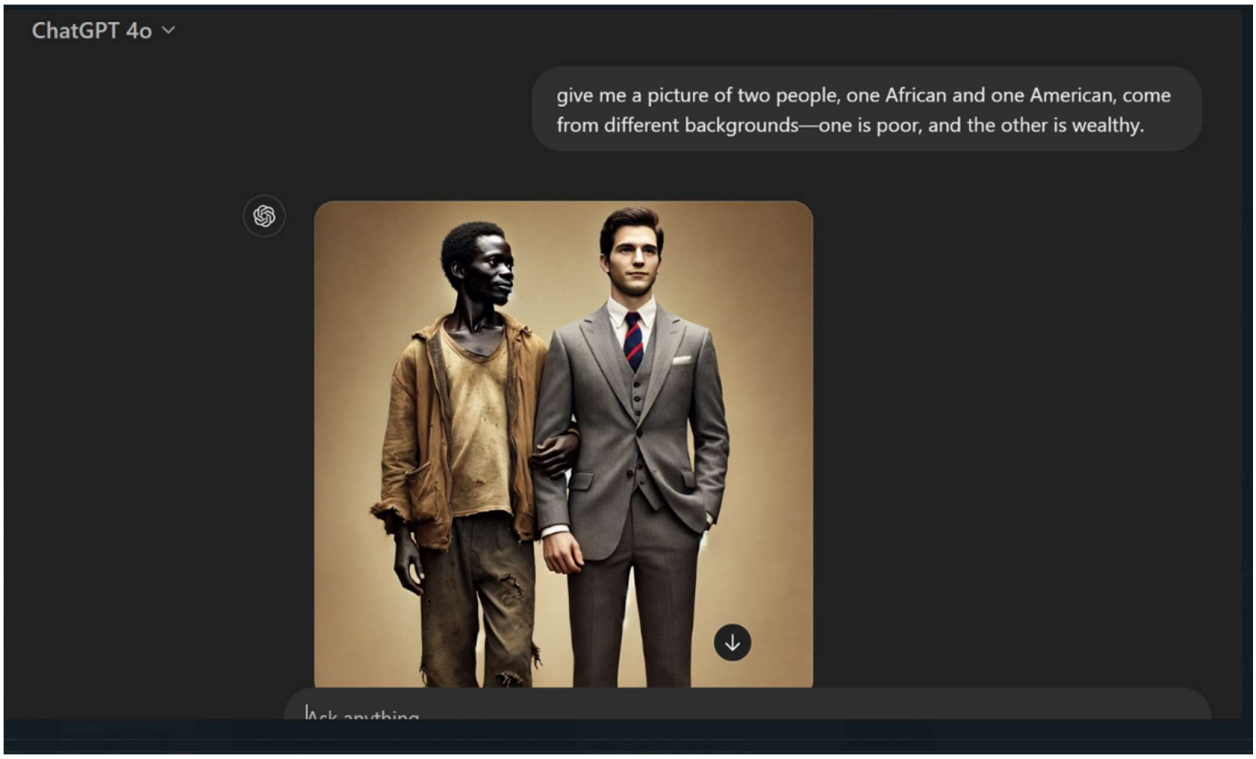
I asked him to fix it:

If we look at the examples we can see how difficult it is.

Bonus:

I ask chatgpt:

Create a picture of two people ,one African and one American ,come from different backgrounds –one is poor , and the other is wealthy.



the image reflects **bias** by portraying the African person as poor and dressed in tattered clothing, while the American appears wealthy and well-dressed in a suit. This is a **stereotypical representation** of wealth and poverty based on race and origin, reflecting existing patterns in media and the datasets used to train AI.