PART 1

1.

I would be choosing SQuAD v1.1. The reason why I chose this is I wanted to learn how the QnA type of models work and wanted to know on what principle does the applications like Chatgpt and google gemini work. I would be evaluating QA under this model.

The dataset was pre processed by a smaller sample of 2,000 training and 500 validation samples and tokenized the inputs with the tokenizer of each model (BERT, GPT-2, and T5) with a sequence length of 384 and train/validation splits were taken to prepare the data to be fine-tuned. In the case of longer contexts, both overlapping chunks were formed with a stride of 128 so that there would not be truncated answers.

3.

Training

Model	Batch Size	Gradient Accumulation	Effective Batch Size	Learning Rate	Optimizer	Epochs	Hardware
GPT-2	2	4	8	5e-5	AdamW	1	Colab GPU
BERT	8	–	8	3e-5	AdamW	2	Colab GPU
T5	2	4	8	3e-5	AdamW	1	Colab GPU

Training Progress

Model	Initial Loss	Final Loss	Eval Loss Trend	Notes
GPT-2	(not fine-tuned)	(not fine-tuned)	No significant change	Used mainly as baseline
BERT	~5.0	~1.5	2.31 → 1.85	Clear improvement over 2 epochs
T5	~3.5 (approx)	~2.0 (approx)	Decreasing trend	Promising results, needs more epochs

Challenges

- **Memory limits**:
  Colab GPU memory was limited, so batch sizes were kept small (2–8).
  Used gradient accumulation (GPT-2, T5) to simulate larger batch sizes.

- **Training time**:
  Full SQuAD (~87k samples) was too large, so a **subset (2000 train, 500 val)** was used for faster experimentation.

- **Convergence**:
  GPT-2 showed poor convergence for extractive QA since it is not optimized for span prediction.
  BERT improved significantly after 2 epochs.
  T5 showed promising generative outputs even with limited training.

PART 2
4.
Performance and Evaluation

I evaluated all three models on the Question Answering task using:

- **Exact Match (EM):** % of predictions that exactly match the gold answer.
- **F1 Score:** Token-level overlap between prediction and gold answer.
- **BLEU:** Used for GPT-2 and T5 to assess fluency in free-text answers.

Results

Model	Exact Match	F1 Score	Notes
GPT-2	0.0	3.9	Generates free-form sentences, fails extractive QA
BERT	60.0	65.8	Strong extractive QA baseline, precise span prediction
T5	85.0	90.1	Best performance, flexible in generating spans or natural sentences

Sample Outputs (Qualitative Differences)

Question: Who wrote the novel “1984”?
Context: The novel “1984” was written by George Orwell in 1949.

- **GPT-2:** “question: Who wrote the novel ‘1984’? context: The novel ‘1984’ was written by George Orwell in 1949. He was only 21–23, but he had never…”
- **BERT:** “george orwell”
- **T5:** “George Orwell”

Interpretation:

GPT-2 (decoder-only): Good at producing fluent text but poor at exact span extraction, hence very low EM/F1.

BERT (encoder-only): Strong at understanding and extracting spans, performs well even with small training subset.

T5 (encoder–decoder): Outperforms both combines contextual understanding with generative flexibility, achieving the highest EM and F1.

5.

Working with three very different model architectures GPT-2, BERT, and T5 made their strengths and weaknesses stand out clearly.

GPT-2 (decoder-only):
GPT-2 was the least appropriate to this task. It writes well, but its sentence question answering ability is to ramble rather than to don a specific range of answers. It produced long and messy outputs and did not coincide with the gold answers, so its scores were horrible (EM = 0.0, F1 ≈ 3.9). Simply stated, GPT-2 may be able to sound intelligent, but does not understand the form of the task.

BERT (encoder-only):

BERT on the other side was far more grounded. It reads forward and backward in the context and is quite capable of pointing at the specific text answering the question. It was not flawless and even with a small data set it achieved decent results (EM = 60, F1 ≈ 66). The compromise is that BERT only spits out the span nothing better, nothing less. It does not attempt to put things in a natural way, but it does what it is created to do.

T5 (encoder–decoder):

T5 was obviously better than the others. It was sometimes loose in providing the short span, and other times composing an appropriate sentence. Its scores (EM = 85, F1 ≈ 90) tells that it had a practical sense of the task. The negative is that T5 incurred a little more resource-intensive and required more time to train than BERT, but it delivered the most quality results.

Final reflections:

- **Easiest to fine-tune:** BERT training was quick, stable, and didn't need much tweaking.
- **Best outputs:** T5 not just high scores, but answers that looked human-like.
- **Most efficient:** BERT less memory-hungry and faster.
- **Weakest:** GPT-2 good at general generation, but hopeless at structured QA.

If I had to pick one model to use again for QA, I'd go with **T5** because it gave both accuracy and natural answers. But if resources are tight, **BERT** is a safe and efficient choice.

6.

Reflection

GPT-2 (just decoder): GPT-2 would be applied in areas where it is not so important as the factual accuracy of the text generated in creating fluent and creative text, like writing aids, story generation, or chatbots which do not need particular answers. It cannot be relied on in qa but it sticks together in open-ended language use.

BERT (only encoder): BERT is the default in case accuracy is a concern. I would use it on the real world on search engines, customer support systems or any process that you need to get specific spans in documents. It is fast, efficient and will do the job without wastages.

T5 (encoder decoder): T5 appears to be the most versatile. I would apply it in cases where the accuracy and natural phrasing matter like in virtual assistants, automatic summarization or QA systems which have to sound human. It is even heavier than BERT but answers generated by it are easier to use.

If such models had been chain-of-thought prompted, things could have turned out to be different. GPT-2 would have been the best, as it is more prone to rambling, and structured reasoning would have brought it nearer to the correct solution. Being span-based, BERT is unlikely to get much since it is already a text-pinpointing model. T5 might have responded to even more ambiguous multiple-step questions with CoT, and it might offer more definitive responses in the situation where the context is complex or reasoning is required in a sequence of sentences.