Rubric for Q2 (Encoder) and Q3 (DIY Transformer)

Q2 Encoder (40 points)

Q2.1 Task 1 (8 points)

- +8: Correct
- +2: Code computes embeddings for all training examples (starting from index 1)
- +2: Code correctly identifies the most similar review (maximum cosine similarity)
- +2: Code correctly identifies the most dissimilar review (minimum cosine similarity)
- +2: Code correctly identifies the most similar review again (maximum cosine similarity)

Q2.2 Task 2 (8 points)

- +8: Correct
- +3: `SentimentBert.init` includes a classifier layer (`nn.Linear(768, 1)` or `nn.Linear(768, 2)`)
- +3: `SentimentBert.forward` extracts [CLS] representation from BERT output (first token)
- +2: `SentimentBert.forward` applies classifier and returns logits

Q2.3 Task 3 (12 points)

- +12: Correct
- +2: Training loop sets model to training mode and iterates through batches
- +2: Code moves `input_ids`, `attention_mask`, and `labels` to device (GPU/CPU)
- +2: Forward pass uses both `input_ids` and `attention_mask`
- +2: Loss function is appropriate for binary classification (`BCEWithLogitsLoss` for 1 output OR `CrossEntropyLoss` for 2 outputs)
- +2: Training prints training loss and validation accuracy each epoch
- +2: Model achieves at least 80% validation accuracy

Q2.4 Task 4 (12 points)

- +12: Correct
- +2: Code computes and prints test accuracy on held-out test data
- +2: Code identifies and prints 5 incorrect predictions
- +2: For each incorrect example, output includes review text, true label, and predicted label
- +3: Written reflection discusses the qualitative examples and whether labels are ambiguous

- +3: Written reflection explains why qualitative evaluation is important alongside quantitative metrics for real-world NLP deployment

---

Q3 DIY Transformer (60 points)

Q3.1 Task 1 (24 points)

- +24: Correct
- +2: `TransformerBlock.__init__` includes `nn.MultiheadAttention` with correct `embed_dim` and `num_heads`
- +2: Includes two `nn.LayerNorm` layers
- +2: Includes feedforward network (MLP) with hidden layer and activation
- +2: `forward` applies multi-head attention with `attn_mask`
- +2: First residual connection and layer normalization after attention
- +2: Second residual connection and layer normalization after feedforward network
- +2: `Transformer.__init__` includes word embedding layer
- +2: Includes positional embedding layer
- +2: Includes multiple `TransformerBlock` instances (via `nn.ModuleList` or equivalent)
- +2: Includes final layer normalization and linear projection to vocab size
- +3: `forward` adds word and positional embeddings
- +3: `forward` creates causal attention mask (upper triangular with -inf values using `torch.triu`)

Q3.2 Task 2 (18 points)

- +18: Correct
- +2: Model hyperparameters meet specifications (heads, blocks, `d_embed`)
- +2: Training loop iterates through dataset and computes forward pass
- +2: Computes loss with cross-entropy
- +2: Backward pass updates model parameters using optimizer
- +2: Prints average loss per epoch
- +2: Prints accuracy per epoch
- +3: Achieves at least 70% training accuracy
- +3: Reports total number of model parameters with calculation

Q3.3 Task 3 (18 points)

- +18: Correct
- +2: `generate` tokenizes `start_text` and handles `<eos>` token
- +2: Applies temperature scaling (logits / temperature)
- +2: Applies softmax and samples next token

- +2: Stops when `<eos>` or `max_tokens` is reached
- +2: Returns decoded string output
- +2: Demonstrates 3 temperature experiments (0.5, 1.0, 2.0) with same prompt
- +2: Demonstrates at least 3 training-data prompts (temp=1.0)
- +2: Demonstrates at least 3 novel prompts (temp=1.0)
- +1: Written discussion explains effect of temperature on generation quality/diversity
- +2: Written discussion compares training-data vs novel prompt generations and explains observed differences