Ex1 - Yael Belfer

Question 1

Dataset 1: QA-SRL (Question-Answer driven Semantic Role Labeling)

This dataset turns semantic role labeling into a QA task, where questions like “Who did something?” or “What was done?” are asked about verbs in a sentence. It tests intrinsic language understanding because the model has to understand the meaning and structure of the sentence to answer correctly. In addition, the task is not straightforward such as translation tasks are (it is not a real world application).

Dataset 2: QA-ZRE (Question Answering for Zero-shot Relation Extraction)

This dataset reformulates relation extraction as QA: given a sentence and a question like "Where was [person] born?", the model extracts the answer. It tests whether the model understands semantic relations — an intrinsic linguistic ability.

Dataset 3: Quoref (Question Understanding of Coreference)

Quoref is a QA dataset that checks if a model can figure out when different words or phrases refer to the same thing in a passage (coreference). This tests intrinsic language understanding because the model has to really understand how the sentence is built and how the ideas connect — it’s not just about finding matching words or solving a real-world task.

Question 2

a)

FlashAttention

Description: FlashAttention is a fast and memory-efficient way to compute attention by combining all attention operations (matmul, softmax, etc.) into a single GPU kernel. This avoids repeated memory access to the slower GPU memory (HBM) by keeping data in faster memory (SRAM).

Advantages:

- Significantly reduces memory bandwidth usage.
- Improves runtime performance.
- Integrates easily with PyTorch/transformers.

Computational bottlenecks:

- Relies on limited fast memory (SRAM).
- Can’t easily extract intermediate attention values because it’s merged.

Could the method be parallelized?:

Yes, it runs efficiently on GPUs and supports parallelism across attention heads (FlashAttention-2 better parallelizes the attention over GPUs).

FlexAttention

Description**:**
FlexAttention is a more flexible version of FlashAttention. It allows token-wise modifications to the attention score function, enabling dynamic patterns like sliding window attention. It still benefits from fast fused kernel execution. Instead of calculating the attention regularly - they added an option to implement score function that works per entry (score_mod)

Advantages**:**
• Enables custom attention patterns (e.g., sliding window).
• Maintains high speed with efficient GPU kernel usage.
• Offers more flexibility in attention computation.

Computational bottlenecks:
• Slightly higher complexity due to per-token score modifications.
• Needs tuning for specific tasks to remain efficient.
Could the method be parallelized?
Yes, it supports GPU-level parallelism and runs efficiently across multiple heads.

RingAttention

Description**:** RingAttention is designed for very long sequences. It distributes the attention computation across multiple GPUs by passing blocks of keys and values between them in a ring-like pattern, allowing each GPU to compute part of the attention.

Advantages**:**
• Handles arbitrarily long sequences.
• Makes full use of multi-GPU systems.
• Parallelizes both attention computation and communication.

Computational bottlenecks:
• Communication overhead between GPUs.
• Synchronization costs across devices.
Could the method be parallelized?
Yes, it's specifically designed to parallelize attention over multiple GPUs.

b) If I had a single GPU (but a powerful one with large memory capacity) and needed to perform a task that involves reasoning, I would use FlashAttention-2. The reason is that Reasoning tasks often involve long sequences, deep dependencies, and require precise attention over relevant information.FlashAttention-2 is highly optimized for memory and compute efficiency on a single GPU, letting you fit more context and go deeper without running out of memory.
Compared to FlexAttention: FlashAttention-2 is faster and more optimized for standard attention. FlexAttention is more flexible, but that flexibility adds overhead you don’t always need for reasoning tasks.

Compared to RingAttention: RingAttention is designed for multi-GPU setups. On a single GPU, it doesn't help and may even add unnecessary complexity.

Question 2.1

2.

Yes. In my experiments, the configuration that achieved the highest validation accuracy also yielded the highest test accuracy. Specifically, the model trained with a learning rate of 2e-5, batch size of 16, and 3 epochs reached a test accuracy of 0.8197, outperforming the other configurations:

- `lr2e-05_bs16_ep3` → **0.8197**
- `lr0.001_bs8_ep2` → 0.6649
- `lr1e-05_bs32_ep1` → 0.7304

This suggests that my validation set was a good proxy for generalization, and the hyperparameter tuning process successfully identified the most robust model.

Qualitative Analysis

To better understand what distinguished the best and worst models, I manually examined prediction outputs from the `predictions.txt` file.

- The **best model** (`lr2e-05_bs16_ep3`) was more consistent in correctly identifying **paraphrases** that involved **rewording** or **syntactic variation**, such as passive-to-active conversions or synonym replacements.
- The **worst model** (`lr0.001_bs8_ep2`) frequently failed on such cases and tended to default to predicting "not paraphrase" in borderline examples, suggesting underfitting or unstable training due to too high a learning rate.

Example 1 (Correctly predicted by best model, failed by worst):

Sentence 1: "A tropical storm rapidly developed in the Gulf of Mexico Sunday..."
Sentence 2: "A tropical storm rapidly developed in the Gulf of Mexico on Sunday..."
True label: 1
Best model: 1
Worst model: 0

Here, the worst model likely missed the strong lexical and syntactic overlap due to weak generalization.

Example 2 (Correct paraphrase prediction by best model):

Sentence 1: "Air Commodore Quaife said the Hornets remained on three-minute alert..."
Sentence 2: "Air Commodore John Quaife said the security operation was unprecedented."
True label: 0
Best model: 0

In contrast, the best model correctly differentiated these as not paraphrases despite surface-level similarities in structure and entity mentions.

Summary

This analysis confirms that proper hyperparameter tuning significantly impacts performance, and validation accuracy can be a strong indicator of real-world generalization when the validation split is representative. The best model demonstrated stronger semantic understanding and more robust decision-making on difficult examples.