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)

2. Inference-Time Scaling

(a) Methods

1. Chain-of-Thought (CoT) Prompting

- Description:
  The model is prompted to "think step by step" before answering. The chain length controls the amount of reasoning at inference time.

- Advantages:
  - Boosts performance on reasoning-heavy tasks.
  - More interpretable outputs.
  - No need to change or retrain the model.

- **Computational Bottlenecks:**

    - Longer outputs → more tokens → higher latency and memory usage.
    - Generation is sequential (can't parallelize within one chain).
- **Parallelizability:**
    - Yes across samples (batching).
    - No within a single sample (sequential token generation).

2. Self-Consistency

- Description:
  Sample multiple CoT outputs and take the majority/most common final answer.
- Advantages:
    - Reduces answer variance.
    - More accurate and robust than single-chain output.
- Computational Bottlenecks:
    - Needs multiple generations per input → expensive.
    - High memory use.
- Parallelizability:
    - Yes, but mostly useful with multi-GPU setups.

3. Verifiers

- Description:
  Use a separate model or rules to check and filter generations (e.g., logic checkers or test sets).
- Advantages:
    - Helps reject wrong or illogical outputs.
    - Can boost performance when combined with other methods.

- Computational Bottlenecks:
    - Adds extra compute for evaluating each generation.
    - Depends on verifier complexity.
- Parallelizability:
    - Yes, but again, benefits most from multi-GPU setups.

(b) My Choice

I would choose Chain-of-Thought prompting.

- Why:
    - It improves reasoning without requiring multiple samples or extra models.
    - Works well on a single GPU — no need for multi-GPU parallelism.
    - I can scale the quality by adjusting the length of the reasoning chain, which my GPU's memory can handle.
    - Simple to implement, efficient, and gets good results on reasoning tasks.

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.