Introduction to Large Language Models
Assignment- 8

Number of questions: 10 **Total mark: 10 X 1 = 10**

QUESTION 1: [1 mark]

Which factors influence the effectiveness of instruction tuning?

a) The number of instruction templates used in training.
b) The tokenization algorithm used by the model.
c) The diversity of tasks in the fine-tuning dataset.
d) The order in which tasks are presented during fine-tuning.

Correct Answers: a, c, d

Explanation:

1. **Number of instruction templates (a):** Having multiple, varied instruction templates helps the model learn how to follow instructions in different styles and formats. This increases the robustness of the model's instruction-following capability.

2. **Diversity of tasks in the fine-tuning dataset (c):** Exposing the model to a broad spectrum of tasks (e.g., classification, summarization, translation) teaches it to generalize better. If the dataset lacks diversity, the model risks overfitting to a narrow range of instructions.

3. **Order in which tasks are presented (d):** The sequence of tasks during multi-task or sequential fine-tuning can affect how well the model retains and integrates learned instructions. Certain curricula strategies (i.e., scheduling tasks in a deliberate sequence) often yield better performance.

4. **Why not the tokenization algorithm (b)?** While tokenization can impact model performance overall, it is not typically considered a primary factor in the effectiveness of instruction tuning itself. It plays a role in how text is broken down but is less critical than the factors in (a), (c), and (d).

QUESTION 2: [1 mark]

What are key challenges of soft prompts in prompt-based learning?

a) Forward pass with them is computationally inefficient compared to that with hard prompts.
b) They require additional training, unlike discrete prompts.
c) They cannot be interpreted or used effectively by non-expert users.
d) They require specialized architectures that differ from standard transformers.

Correct Answers: b, c

Explanation:

1. **They require additional training (b):** Soft prompts are continuous embeddings learned during training. Unlike hard (discrete) prompts, which are just tokens inserted into the text, soft prompts require gradient-based optimization. This necessitates a training procedure rather than a simple manual prompt.

2. **They cannot be interpreted or used effectively by non-expert users (c):** Since soft prompts are embedding vectors (continuous parameters), humans cannot directly read or interpret them. They are not as transparent or easily modifiable as typed-out text prompts.

3. **Why not (a) or (d)?**

   (a) Soft prompts do not necessarily result in a *computationally inefficient* forward pass compared to normal token embeddings. The computational overhead is minor and similar to standard embedding lookups.

   (d) Soft prompts generally *do not* require a specialized architecture; they can be applied to standard transformer-based models.

QUESTION 3: [1 mark]

Which statement best describes the impact of fine-tuning versus prompting in LLMs?

a) Fine-tuning is always superior to prompting in generalization tasks.
b) Prompting requires gradient updates, while fine-tuning does not.
c) Fine-tuning modifies the model weights permanently, while prompting does not.
d) Prompting performs better on in-domain tasks compared to fine-tuning.

Correct Answer: c

Explanation:

- **Fine-tuning modifies model weights permanently (c):** During fine-tuning, the model's parameters are updated, leading to a permanent change in its internal weights. In contrast, prompting relies on creating or inserting instructions (hard prompts) or learned embeddings (soft prompts) without altering the core model parameters.

- **Why not a, b, or d?**

  (a) Fine-tuning is not always superior to prompting; performance can vary based on task, data availability, and use case.

(b) Prompting usually does not require gradient updates (unless you are using soft prompts or another trainable method of prompting). Fine-tuning does require gradient updates.

(d) Prompting does not necessarily outperform fine-tuning on in-domain tasks; in some scenarios, a well-fine-tuned model can achieve higher in-domain accuracy.

QUESTION 4: [1 mark]

Which of the following aspects of the model outputs are captured by POSIX?

a) Diversity in the responses to intent-preserving prompt variations
b) Entropy of the distribution of response frequencies
c) Time required to generate responses for intent-preserving prompt variations
d) Variance in the log-likelihood of the same response for different input prompt variations

Correct Answer: a, b, d

Explanation:

1. **Diversity in responses (a):** POSIX aims to measure how many different valid responses the model produces under prompt variations that preserve the same intent.

2. **Entropy of response frequencies (b):** Entropy reflects how spread out or uncertain the model's responses are. POSIX takes into account how often each distinct response occurs, capturing response variability.

3. **Variance in log-likelihood under different prompts (d):** By measuring the log-likelihood of a particular response across varied prompts, POSIX can quantify how sensitive the model is to small changes in prompt phrasing.

4. **Why not (c)?**

   The time required for inference is not directly part of POSIX's metric. POSIX primarily focuses on response diversity and sensitivity to prompt variations, not computational speed.

QUESTION 5: [1 mark]

Which key mechanism makes Tree-of-Thought (ToT) prompting more effective than Chain-of-Thought (CoT)?

a) ToT uses reinforcement learning for better generalization.
b) ToT allows backtracking to explore multiple reasoning paths.
c) ToT reduces hallucination by using domain-specific heuristics.
d) ToT eliminates the need for manual prompt engineering.

Correct Answer: b

Explanation:

- **ToT allows backtracking (b):** Tree-of-Thought prompting structures the reasoning process in a tree-shaped manner, exploring multiple branches of thought. It can backtrack and test different solution paths, which helps uncover correct reasoning steps even if some initial paths were flawed.

- **Why not a, c, or d?**

  (a) While some implementations might use reinforcement learning, it is not the defining reason for ToT's effectiveness.

  (c) ToT does not inherently rely on domain-specific heuristics to reduce hallucination—rather, it systematically explores more possible reasoning paths.

  (d) ToT still benefits from careful prompt design for best results.

QUESTION 6: [1 mark]

What is a key limitation of measuring accuracy alone when evaluating LLMs?

a) Accuracy is always correlated with model size.
b) Accuracy cannot be measured on open-ended tasks.
c) Accuracy is independent of the training dataset size.
d) Accuracy does not account for prompt sensitivity.

Correct Answer: d

Explanation:

- **Accuracy does not account for prompt sensitivity (d):** Large Language Models can produce significantly different responses depending on how the prompt is phrased. Simply measuring accuracy on a set of prompts does not tell us whether the model's performance is stable across different prompt variations.

- **Why not a, b, or c?**

  (a) Accuracy is not always correlated with model size.

(b) Accuracy can be measured (in some form) even on open-ended tasks if appropriate metrics or reference answers are defined.

(c) Accuracy can be influenced by the training data size and quality. It is not truly independent of dataset size.

QUESTION 7: [1 mark]

Why is instruction tuning not sufficient for aligning large language models?

a) It does not generalize to unseen tasks.
b) It cannot prevent models from generating undesired responses.
c) It reduces model performance on downstream tasks.
d) It makes models less capable of learning from new data.

Correct Answer: b

Explanation:

- **It cannot prevent models from generating undesired responses (b):** While instruction tuning can help a model better follow user instructions, it does not fully address safe or ethical generation. A model might still produce harmful, biased, or factually incorrect outputs if not further aligned with techniques such as reinforcement learning from human feedback (RLHF) or safety guardrails.

- **Why not a, c, or d?**

  (a) Properly designed instruction tuning can help models generalize to new tasks, though not perfectly.

  (c) Instruction tuning does not inherently reduce performance on downstream tasks—often it aims to improve it.

  (d) Instruction tuning does not necessarily hinder a model's ability to learn from new data.

QUESTION 8: [1 mark]

Why is KL divergence minimized in regularized reward maximization?

a) To maximize the probability of generating high-reward responses.
b) To make training more computationally efficient.
c) To prevent the amplification of bias in training data.
d) To ensure models do not diverge too far from the reference model.

Correct Answer: d

Explanation:

- **Ensuring the model does not diverge too far (d):** Minimizing KL divergence between the updated policy (the newly optimized model) and a reference policy (the base model) constrains the model's changes. This helps the model exploit reward signals while retaining behaviours learned previously, avoiding catastrophic deviations that degrade performance or cause instability.

- **Why not a, b, or c?**

  (a) While reward maximization is the core goal, the reason to minimize KL is not specifically "to maximize high-reward responses" but to balance staying close to the original policy while moving toward higher reward.

  (b) KL divergence minimization does not necessarily make training more efficient computationally; it's more about controlling how drastically the policy updates.

  (c) Minimizing KL divergence can help with bias issues indirectly, but its primary purpose is to control the distribution shift rather than specifically address data bias.

QUESTION 9: [1 mark]

What is the primary advantage of using the log-derivative trick in REINFORCE?

a) Reducing data requirements
b) Expanding the token vocabulary
c) Simplifying gradient computation
d) Improving sampling diversity

Correct Answer: c

Explanation:

- **Simplifying gradient computation (c):** The log-derivative trick in the REINFORCE algorithm allows one to compute gradients of an expectation without having to differentiate the reward function directly (which might be discrete or non-differentiable). By bringing the log of the model's probability into the gradient expression, one can handle policy gradients in a more straightforward way.

- **Why not a, b, or d?**

  (a) The trick primarily addresses gradient calculation; it does not necessarily reduce how much data is needed.

  (b) It is unrelated to increasing the model's vocabulary.

(d) While it might affect how sampling is done during training, improving diversity is not its primary purpose.

QUESTION 10: [1 mark]

Which method combines reward maximization and minimizing KL divergence?

a) REINFORCE
b) Monte Carlo Approximation
c) Proximal Policy Optimization
d) Constitutional AI

Correct Answer: c

Explanation:

- **Proximal Policy Optimization (PPO) (c):** PPO is an algorithm designed to maximize a reward function while explicitly minimizing the KL divergence between the updated policy and the old policy. By doing so, it ensures that each update does not shift the policy too drastically, leading to more stable learning.

- **Why not a, b, or d?**

  (a) REINFORCE maximizes reward but does not inherently include a KL-based constraint to keep the new policy close to the old policy.

  (b) Monte Carlo Approximation is a technique, not a specific RL algorithm that enforces KL constraints.

  (d) Constitutional AI is a framework for aligning language models using a constitution of principles, not specifically about reward maximization combined with KL minimization in the same way PPO is.
