

- Course Introduction

Video
- Course Overview

Reading
- Specialization Overview

Reading
- Helpful tips for Course Completion

Ungraded Plugin
- Basics of Instruction-Tuning

Video
- Instruction-Tuning with Hugging Face

Video
- Instruction Tuning

Ungraded Plugin
- Instruction Fine-Tuning LLMs

Ungraded App Item
- Best Practices for Instruction-Tuning Large Language Models

Reading
- Reward Modeling: Response Evaluation

Video
- Reward Model Training

Video
- Reward Modeling with Hugging Face

Video
- Reward Modeling & Response Evaluation

Ungraded Plugin
- Lab: Reward Modeling

Ungraded App Item
- Practice Quiz: Instruction-Tuning and Reward Modeling

Practice Assignment
- Summary and Highlights

Reading
- Different Approaches to Instruction-Tuning

Graded Assignment

Best Practices for Instruction-Tuning Large Language Models

Instruction-tuning is a powerful fine-tuning approach that adapts large language models (LLMs) to follow specific instructions more effectively, enhancing their usefulness in practical applications. Below, we outline best practices for optimizing instruction-tuning in LLMs.

Data Selection for Instruction-Tuning

High-quality data is crucial for effective instruction-tuning. The selected data should reflect diverse instructions and responses to help the model generalize

and respond accurately across varied scenarios.

- Diverse Dataset Collection:** Use datasets that cover a wide range of topics, contexts, and instructions. Including different prompt types and response styles helps the model handle a broader set of instructions.
- Balance of Specialized and General Data:** While it's beneficial to include domain-specific instructions, balancing this with general data improves versatility, allowing the model to perform well across various domains.

Optimize Prompt Engineering

Effective prompt engineering enables the model to understand and respond appropriately to different instructions.

- Contextual Prompt Design:** Design prompts that reflect real-world use cases and specific contexts the model might encounter. For instance, instructions could vary in formality, complexity, or specificity, helping the model adapt to different audiences.
- Testing Prompt Variability:** Experiment with different prompts to assess how well the model generalizes to unseen instructions. This helps ensure that the model doesn't overly rely on specific patterns or structures.

Measure Response Consistency

Consistency in response quality is key to creating a reliable model.

- Evaluate Accuracy and Consistency:** Regularly test the model with similar instructions to measure consistency. Consistent and accurate responses to repeated instructions indicate a well-tuned model.
- Monitor Task-Specific Performance:** If the model is tuned for a specialized application, evaluate its performance across task-specific scenarios to ensure consistency within that context.

Limit Overfitting on Instruction Style

Overfitting on specific instruction styles or tones can reduce the model's adaptability.

- Style Variety in Instructions:** Include a variety of tones and structures in the instruction dataset to avoid making the model too reliant on specific formats.
- Balance Precision and Flexibility:** Fine-tune the model to be precise in its responses without limiting its ability to adapt to different instruction types.

This balance helps create a model that is accurate yet flexible in understanding various instructions.

Implement Regular Evaluation Metrics

Regular evaluation of the fine-tuned model ensures it meets the desired quality standards.

- Use Metrics for Instruction Adherence:** Implement metrics that evaluate how closely the model's responses align with provided instructions.