Data Preparation for LLMs

100x

Recap: Fine-Tuning

Recap: Fine-Tuning

Pretrained model for specific tasks

- Pretrained models are large language models (LLMs) that have been trained on vast amounts of general data.
- These models can be further specialized for specific tasks through fine-tuning.

Fine-tuning teaches LLM to understand new patterns in data

- Fine-tuning adapts the pretrained model to specific domains or tasks.
- It allows the model to learn task-specific vocabulary, context, and patterns.
- This process enhances the model's performance on targeted applications.

Data importance in fine-tuning

- While architecture and training process are crucial, data quality and relevance are paramount.
- High-quality, task-specific data is essential for effective fine-tuning.
- The data should represent the intended use case and cover edge cases.

Types of Datasets

Types of Datasets

Supervised Fine-Tuning (SFT) Datasets

- Consist of instruction-output pairs
- Often use synthetic data generated by frontier models
- Format: System prompt + User prompt (instruction) and model output (answer)

Preference Alignment Datasets

- Include an instruction with a chosen answer and a rejected answer
- Used for methods like Direct Preference Optimization (DPO)

Challenges in creating data

Challenges in creating data

- Collecting real-world data can be time-consuming and expensive.
- Ensuring data quality, diversity, and lack of bias is difficult.
- Some domains have limited available data due to privacy or scarcity issues.
- Labeling data accurately often requires domain expertise.

Dataset Formats & Popular Datasets

Stanford Alpaca Dataset

- `alpaca_data.json` contains 52K instruction-following data we used for fine-tuning the Alpaca model. This JSON file is a list of dictionaries, each dictionary contains the following fields:
  - `instruction: str`, describes the task the model should perform. Each of the 52K instructions is unique.
  - `input: str`, optional context or input for the task. For example, when the instruction is "Summarize the following article", the input is the article. Around 40% of the examples have an input.
  - `output: str,` the answer to the instruction as generated by `text-davinci-003.`

Examples of Open SFT Datasets

What is a good SFT dataset?

What is a good SFT dataset?

Accuracy

- Factually correct outputs
- Minimal typos
- Preserve model knowledge integrity

Diversity

- Cover a wide range of topics (use-case dependent)
- Include various writing styles

Complexity

- Include complex tasks that force reasoning
- Examples: chain-of-thought reasoning, summarization, "explain like I'm 5"

Creating SFT Datasets: A Recipe

Creating SFT Datasets: A Recipe

Dataset creation recipe

1. **Combine** relevant open-source datasets
2. **Data deduplication** with exact or fuzzy (MinHash) deduplication
3. **Data quality** with rule-based filtering, reward models, or LLM-as-a-judge techniques
4. **Data exploration** with e.g. Lilac, Nomic Atlas, text-clustering
5. **Data generation** to add missing data based on exploration and/or results on downstream tasks

1. Start with open-source datasets (combine multiple datasets)

2. Data deduplication and Decontamination

This step is crucial to ensure the dataset doesn't contain redundant information and isn't contaminated with data that might be in the test set.

Exact deduplication

- Remove identical samples with data normalization (e.g., convert text to lowercase), hash generation (e.g., create an MD5 or SHA-256 hash for each sample), and duplicate removal.

Fuzzy deduplication

- **MinHash**: Fuzzy deduplication with hashing, sorting, and Jaccard similarity (preferred technique).
- **BLOOM filters**: Fuzzy deduplication with hashing and fixed-size vector.

Decontamination

- Remove samples too close to test sets, using either exact or fuzzy filtering.

3. Data quality evaluation

This step helps in filtering out low-quality or irrelevant data that could negatively impact model training.

Rule-based

- Remove samples based on a list of unwanted words, like refusals and "As an AI assistant" (example).

LLM-as-a-judge

- Colab notebook that provides code to rate outputs with Mixtral-7x8B.

Data Prep Kit

- Framework for data preparation for both code and language, with modules in Python, Ray, and Spark, and a wide range of scale from laptops to data centers.

Argilla

- Open-source data curation platform that allows you to filter and annotate datasets in a collaborative way.

4. Data Generation

This step is used to augment your dataset, especially if the initial dataset is small or lacks diversity.

Augmentoolkit

- Framework to convert raw text into datasets using open-source and closed-source models.

Distilabel:

- General-purpose framework that can generate and augment data (SFT, DPO) with techniques like UltraFeedback and DEITA.

5. Data Exploration

This step helps in understanding the composition and characteristics of your dataset using topic clustering and visualization, which can inform further refinement or generation steps.

Nomic Atlas:

- Interact with instructed data to find insights and store embeddings.

6. Iterate: Use insights to generate more data and repeat the process

Best Practices

Best Practices

- Tailor dataset complexity to your use case (e.g., summarization vs. general-purpose)
- For general fine-tuning, aim for topic and style diversity
- Use data quality filters to remove low-quality samples
- Iterate on your dataset based on exploration and analysis