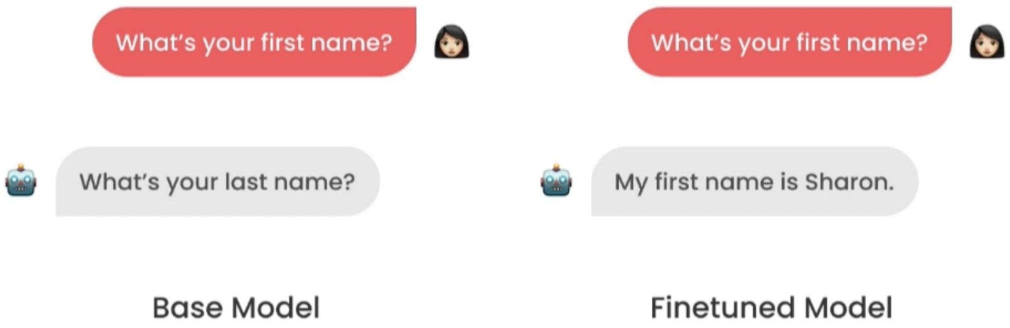
Fine Tuning LLMs

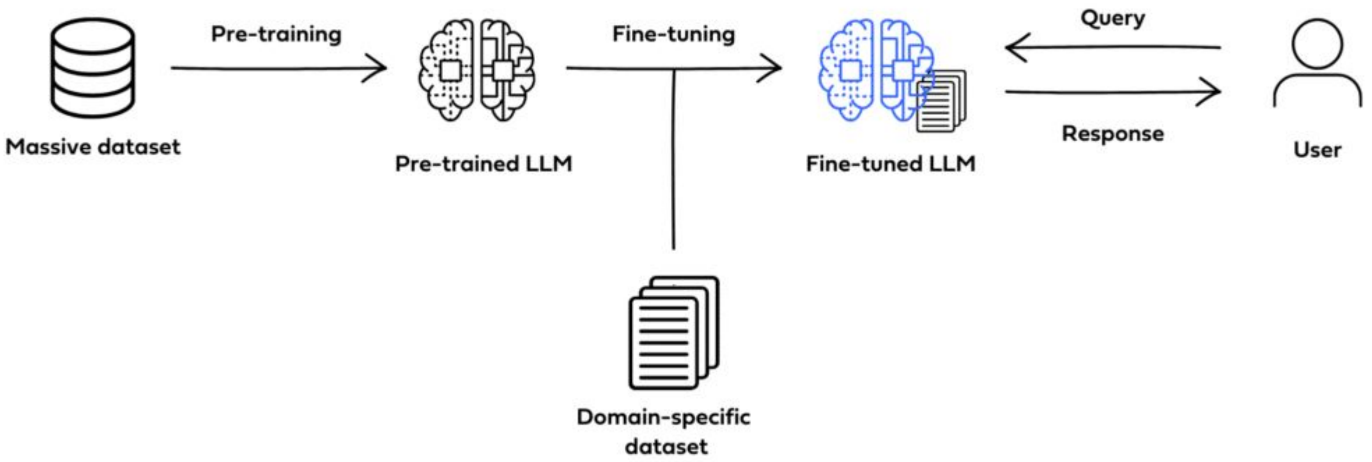
Spurthi Setty

Why should you finetune



- More Consistent Outputs
- Customize models for specific use cases
- Reduces Hallucinations
- Eliminates need of training a model from scratch

Fine Tuning Architecture



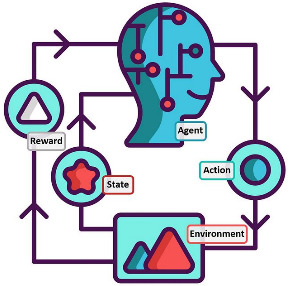
Fine Tuning Methods

- Self-Supervised Learning
  - Unlabeled Data → more scalable
  - Model will mimic the style of the text

- Supervised Learning
  - Training data consists of inputs and outputs
  - Can train for a specific task via instruction fine tuning
    - Ex. Translation, summarization, question-answering

Input	Output

- Reinforcement Learning with Human Feedback
  - Develop a reward model with model outputs and human ratings
  - Proximal Policy Optimization (PPO):



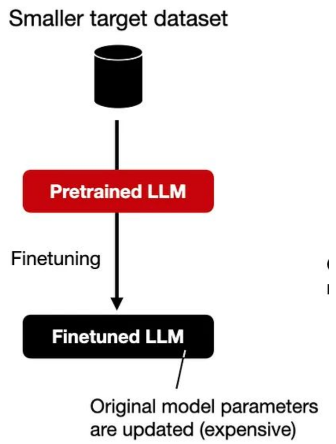
Dataset Creation and Processing

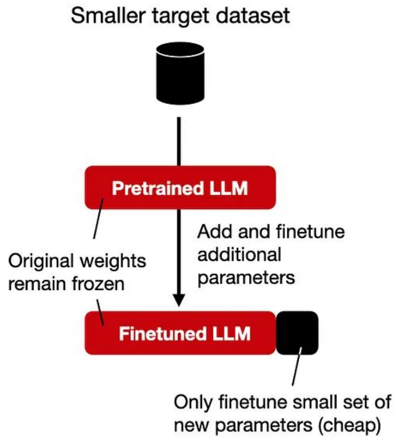
- Each LLM requires a specific format for the training data
  - Ex. Stanford Alpaca format for LLAMA-2
  - GPT requires a specific data format specified in their docs
  - Usually in a json or jsonl format
- High quality data essential for better performance
- Tokenize the formatted data
  - Converts text to numbers
- Train-Test split

```
sample = """

### Instructions: {instruction}

### Input: {Input}

### Response:

"""
```

Methods for Parameter Fine Tuning

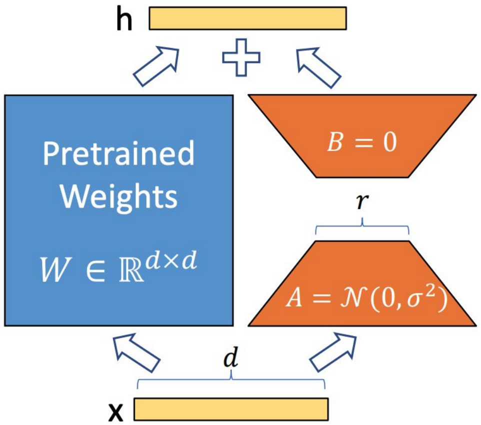
- Full Fine Tuning
  - Adjusts all parameters of the LLM using task-specific data.
  - Computationally expensive
- Transfer Learning
  - Freeze all parameters except for the head of the neural network
  - Only finetune the layers that translate to the output layer
- Parameter Efficient Fine Tuning (PEFT)
  - Freeze all the weights of the base LLM
  - Augment the model with additional parameters and finetune those
  - Less computationally expensive





Low Rank Adaptation (LoRA) for PEFT

- Reduces number of trainable parameters
- Identifies the most crucial parameters for the task at hand and finetunes those
  - approximate the weight matrices of these important layers using low-rank matrices.
- During fine-tuning, only the parameters in the low-rank matrices are updated
- Less chance of overfitting since only a few parameters are updated
- Reduces the computational and memory requirements needed for fine-tuning



Integrating multiple techniques to improve performance

- Prompt Engineering and RAG can be used in conjunction with Fine Tuning
- Use other methods to create optimal training data for Fine Tuning
  - Prompt engineering to optimize system prompts for a specific task
    - Ex. “You are a financial expert.. ”
    - Use prompt engineering to guide outputs to be a certain way
  - Can include context as a part of the input data that is retrieved via RAG
- Use prompt engineering and RAG on a fine tuned model the same way one does on a base model
  - Specialized knowledge from RAG
  - Control and specificity from prompt engineering
- Combination of techniques needed for highly sophisticated, accurate, and efficient AI systems

Thank you!