A Practical Examination of LLMs in Finance Industry

5th Annual Machine Learning in Quantitative Finance Industry
New York, 09/20/2023

Tao Jin
https://www.linkedin.com/in/taojin19/

Agenda

- **Examination of Large Language Model**
- **Examination of LLM Application in Finance Industry**
- **Examination of Building Financial Domain LLM**
- **Operationalizing LLMs for Financial Institution Use-Cases**
- **Challenges**

Examination of Large Language Model - 1. Intro

- **What's Language Model (LM)**

  In general, LM aims to model the generative likelihood of word sequences, so as to predict the probabilities of future (or missing) tokens.

- **Background of Large Language Model (LLM)**

  Four Stages evolve to LLM: Statistical LM, Neural LM, Pre-trained LM and LLM

  Statistical LM : n-gram language models,widely applied to enhance task performance in information retrieval and NLP, However, they often suffer from the curse of dimensionality

  Neural LM: characterize the probability of word sequences by neural networks (RNN)

  Pre-trained LM: pre-trained context-aware word representations are very effective as general-purpose s semantic features, which have largely raised the performance bar of NLP tasks (BiLSTM, BERT)

  LLM: scaling PLM (e.g., scaling model size or data size) often leads to an improved model capacity on downstream tasks, the term "large language models" for these large-sized PLMs

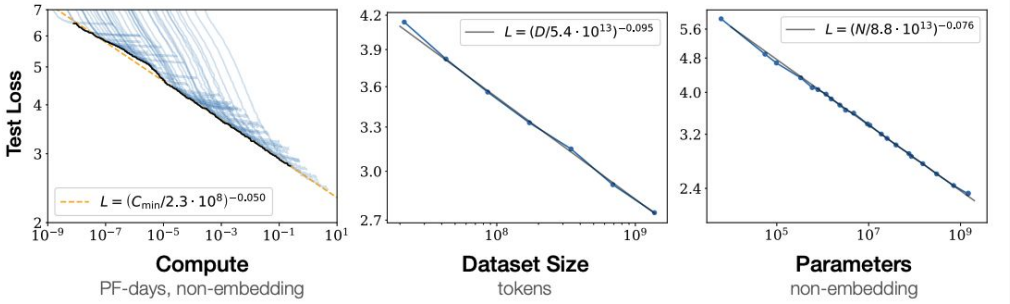
Examination of Large Language Model - 2. Scaling Law

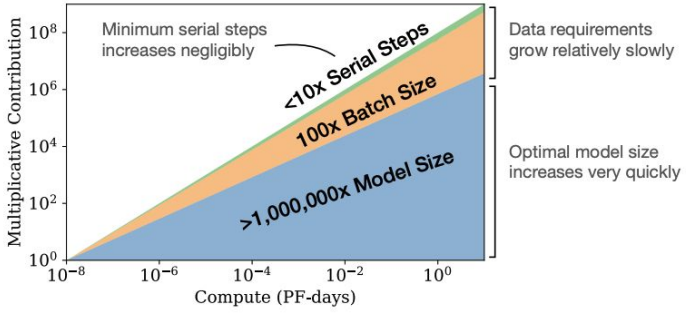
- **Scaling Laws for Large Language Models**

LLMs adopt similar Transformer architectures and pre-training objectives (e.g., language modeling) as small language models. However, LLMs significantly extend the model size (N), data size(D), and total compute (C).
KM Scaling Law (Open AI) vs ***Chinchilla Scaling Law*** (Google Deepmind)

Questions: Given certain quantity of compute, how large of a model should I train in order to get the best possible performance? How to trade off model size vs data size?

GPT-3 20B for 40% internet archive vs 200B for 4% internet archive



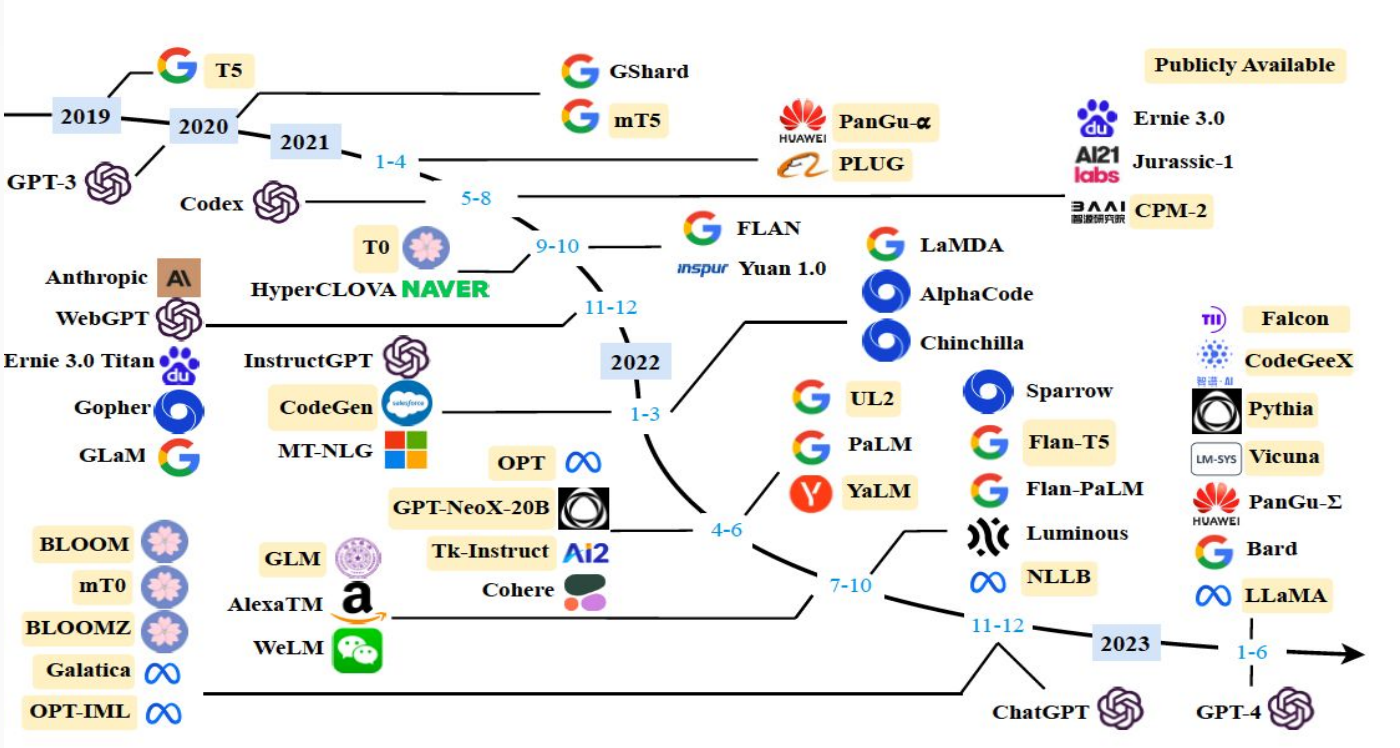


Examination of Large Language Model - 3. Timeline Evolution

- **Timeline of Large Language Models**

A timeline of existing large language models (having a size larger than 10B) in recent years.

.



Examination of Large Language Model - 4. Key Features and Techniques

Key Features for LLMs

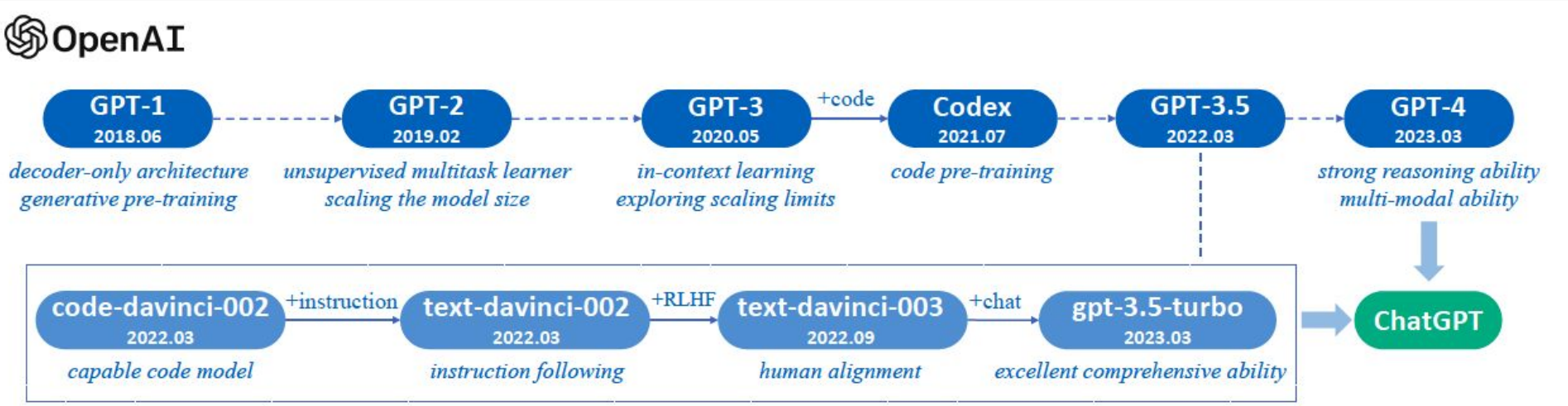
- **In-Context Learning:** providing with a natural language instruction and/or several task demonstrations, it can generate the expected output without requiring additional training or gradient update.
- **Instruction following**: By fine-tuning with a mixture of multi-task datasets formatted via instruction tuning, LLMs perform well on unseen tasks that are also described in the form of instructions
- **Step-by-Step reasoning:** Chain-of-Thought Prompts

Key Techniques for LLMs

- **Scaling:** how to balance the data size, model size and budget
- **Training:** Due to the huge model size, it is very challenging to successfully train a capable LLM
- **Ability eliciting**: After being pre-trained on large-scale corpora, LLMs are endowed with potential abilities as general-purpose task solvers.
- **Alignment tuning:** Since LLMs are trained to capture the data characteristics of pre-training corpora (including both high-quality and low-quality data), they are likely to generate toxic, biased, or even harmful content. It is necessary to align LLMs with human values

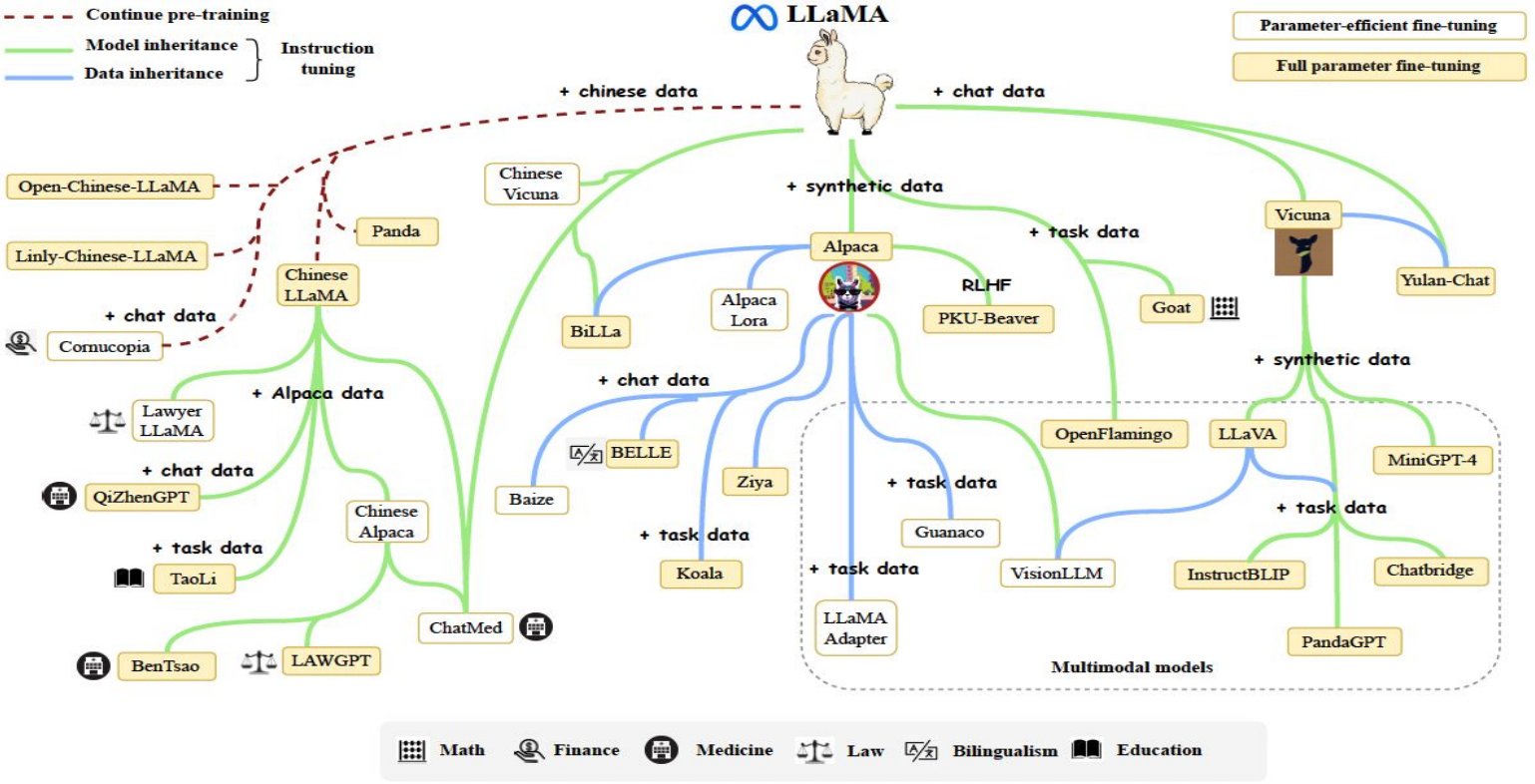
Examination of Large Language Model - 5. GPT Model

- Technical Evaluation GPT Models (Commercial LLM, Closed)



Examination of Large Language Model - 6. Open Source LLM - LLaMA

- Technical Evaluation LLaMA Models (Open Source LLM example)



Examination of LLM Application in Finance Industry - Tasks

Financial NLP tasks: Financial Technology is a large and growing area with NLP technologies having an increasingly important role

- Information Extraction
- Classification
- Sentiment and Topic Analysis
- Q and A for Robot Adviser and Client Support

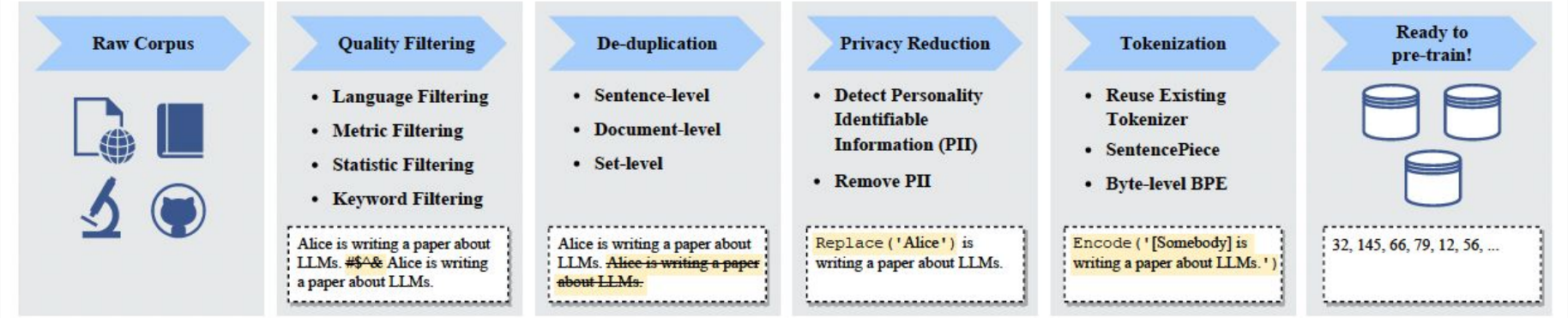
Build Domain-Specific Data sets and Domain-Specific LLMs

Solution: Training from Scratch, Fine-tuning, Prompting and Retrieval-Augment Generation

Examination of LLM Application in Finance Industry - Data Set

- **Build Domain Specific Dataset**

  Domain Specific data will be useful to improve the specific capabilities of LLMs on downstream tasks.



Examination of LLM Application in Finance Industry - BloombergGPT

- **Training from Scratch - BloombergGPT (50B based on BLOOM)**

  Construct a 363 billion token dataset based on Bloomberg's extensive data sources, augmented with 345 billion tokens from general purpose datasets (700 B Token), based on Chinchilla scaling laws.

- **Template Tasks in Financial domain**

Task	Template/Example
Discriminative	

Sentiment Analysis	`{sentence}`
	`Question: what is the sentiment?`
	`Answer: {negative/neutral/positive}`

Aspect Sentiment Analysis	`{sentence}`
	`Question: what is the sentiment on {target}?`
	`Answer: {negative/neutral/positive}`

Binary Classification	`{sentence}`
	`Question: {question}?`
	`Answer: {Yes/No}`

Generative	
NER	Steve Jobs is the CEO of Apple Extract named entity: Steve Jobs (person), Apple (organization)
NER+NED	AAPL stopped using Intel Chips Extract ticker: AAPL, INTC

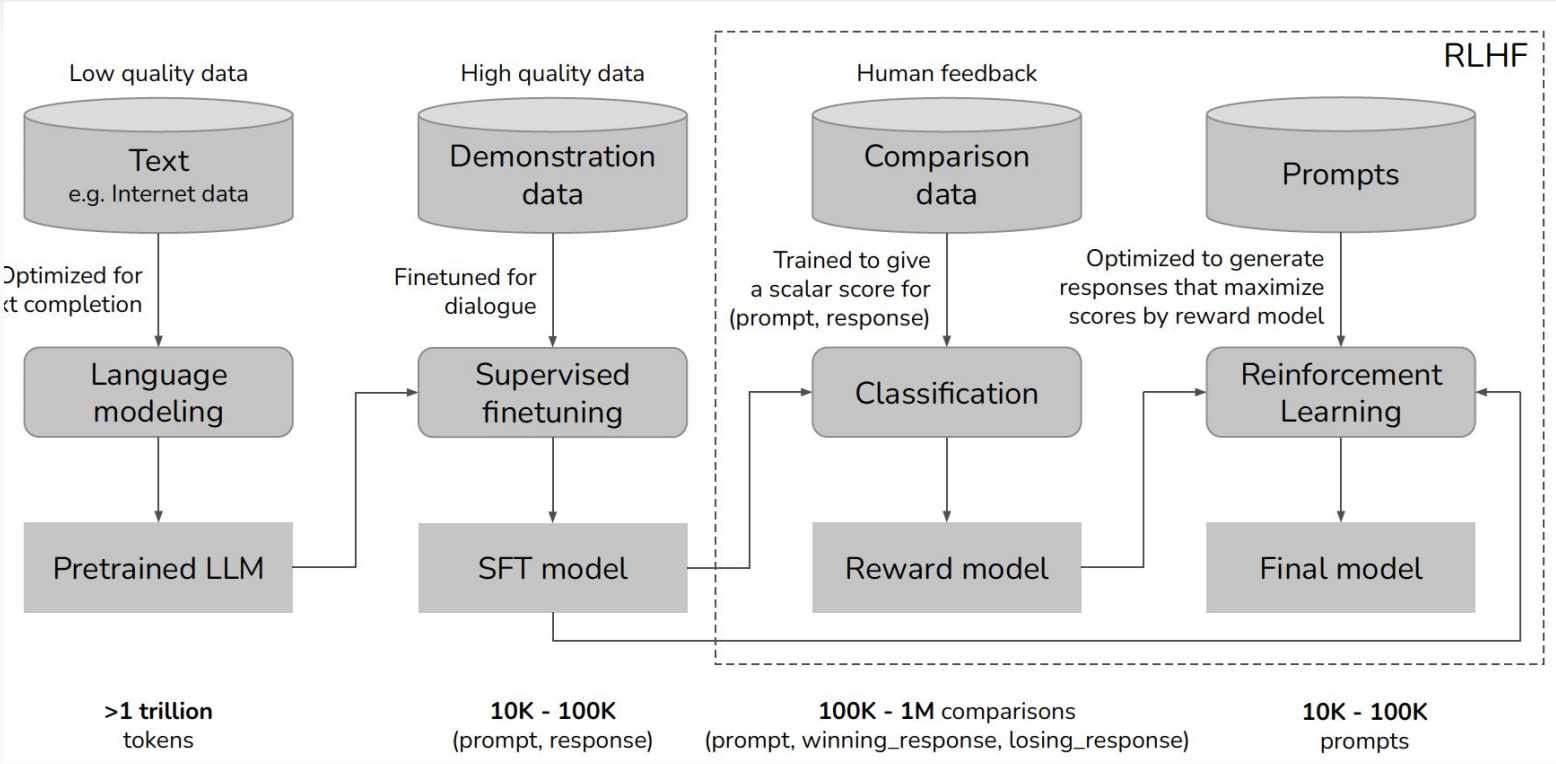
QA	`{context}`
	`Question: {question}?`
	`Answer: {answer}`

Examination of Building Financial Domain LLM - Overview

- Training from Scratch
- Prompting Engineering
- Fine tuning
- RAG

Examination of Building Financial Domain LLM - Training from Scratch

- Training from Scratch



Examination of Building Financial Domain LLM - Training from Scratch

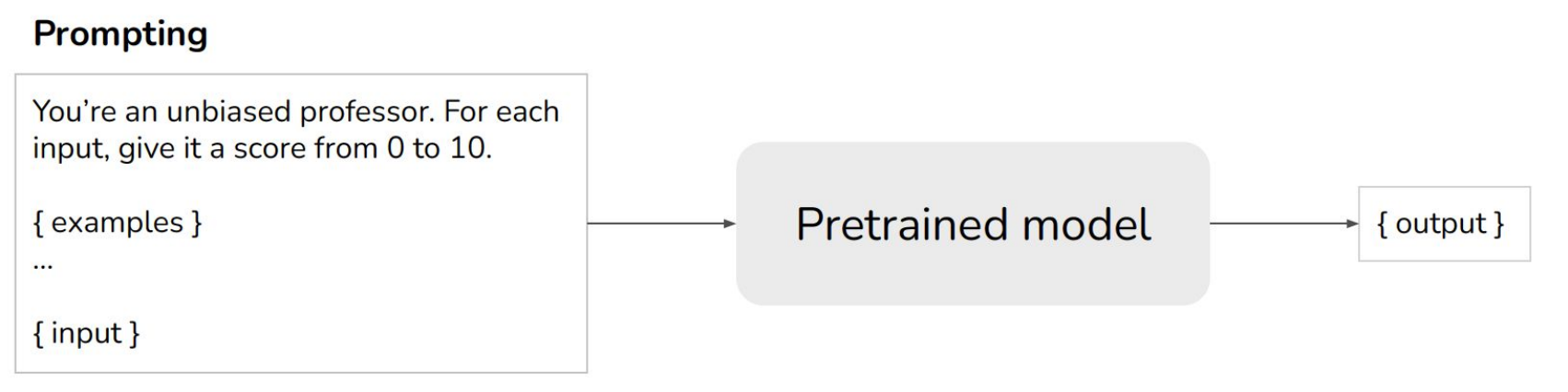
Lessons from RLHF Training

- Reinforcement Learning with Human Feedback (RLHF) is a powerful technique for aligning language models with human preferences. However, the RL stage, Proximal Policy Optimization (PPO), requires over 3x the memory of Supervised Fine-Tuning (SFT), making it infeasible to use for most practitioners.
- Hydra-RLHF is a new method that addresses this challenge by integrating the SFT and Reward models and then dynamically turning LoRA "off" during training.
- Hydra-RLHF reduces the memory usage of PPO to be smaller than SFT while improving alignment across four public benchmarks.
- Hydra-RLHF also reduces the latency per sample of LoRA-PPO by up to 65% while maintaining its performance.

Examination of Building Financial Domain LLM - Prompting

- **Prompting Engineering**

Prompt engineering doesn't change the update the parameters of the model; it's about refining how you talk to the model to get the best out of it.



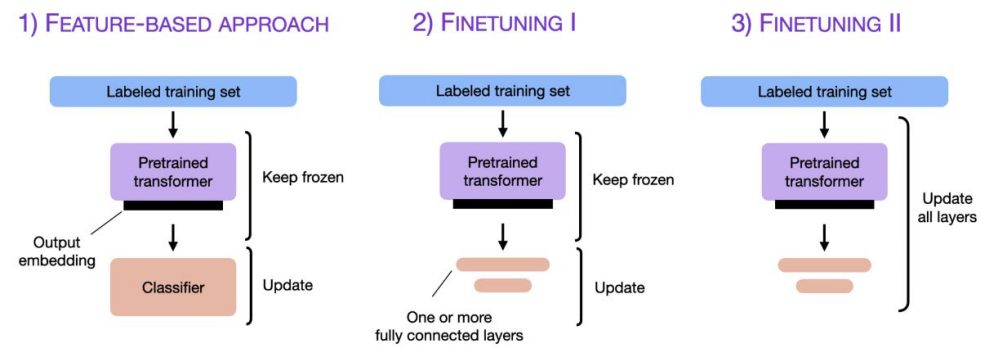
Examination of Building Financial Domain LLM - Fine Tuning

- **Why Fine tuning**

·If we have access to the LLM, adapting and fine-tuning it on a target task using data from a target domain usually leads to superior results.

- **Conventional Fine tuning Approaches**

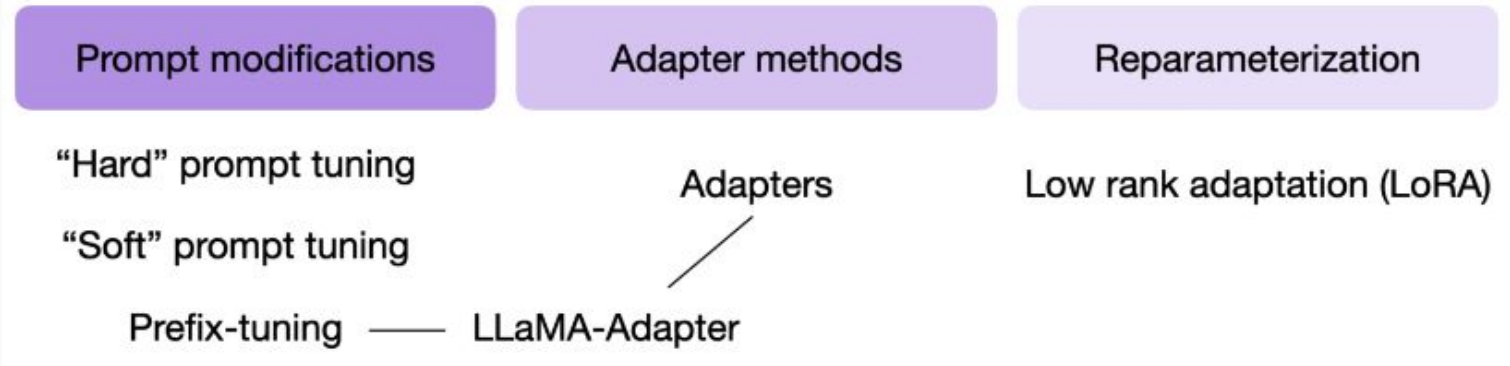
Feature based, Fine Tuning I - updating with output layers, Fine Tuning II - updating all layers



Examination of Building Financial Domain LLM - Fine Tuning

- **Parameter-Efficient Fine Tuning - PEFT**

·The parameter efficient learning fine-tunes a few parameters either by adding new parameters to the model or the existing ones. PEFT methods aim to reduce the computational and memory requirements of fine-tuning LLMs, while still achieving good performance on downstream tasks.



Examination of Building Financial Domain LLM - Fine Tuning

- Prompt Tuning, Prefix Tuning and Adapter Tuning

Method	Description	Advantages	Disadvantages
Prompt tuning	Prepend a trainable prompt to the input text.	Simple to implement, does not require modifications to the LLM architecture.	Not as efficient as prefix tuning or adapter tuning.
Prefix tuning	Prepend a trainable prefix to each layer of the LLM.	More efficient than prompt tuning, can have a more global impact on the model's output.	More difficult to implement than prompt tuning, requires modifications to the LLM architecture.
Adapter tuning	Add trainable modules (adapters) to each layer of the LLM.	Most efficient method in terms of training time and memory requirements, can be added to any LLM architecture.	More difficult to implement than prompt tuning, requires some knowledge of the LLM architecture.

Examination of Building Financial Domain LLM - Fine Tuning

- **Making Weight Updates More Efficient**

·LoRA (Low-Rank Adaptation) and QLoRA (Quantized Low-Rank Adaptation) are two parameter-efficient fine-tuning (PEFT) methods for LLMs.

- **Benefits of using LoRA and QLoRA for LLM fine-tuning**

Reduced computational requirements: significantly reduce the computational time and resources required for fine-tuning LLMs.

Reduced memory footprint: reduce the memory footprint of fine-tuned LLMs, making them easier to deploy on resource-constrained devices.

Good performance: achieve good performance on a variety of downstream tasks, without sacrificing accuracy.

Examination of Building Financial Domain LLM - Fine Tuning

- **Efficient Fine tuning of Quantized LLMs**

LoRA: Input -> Linear layer -> Weight decomposition -> Fine-tuned linear layer -> Output

QLoRA: Input -> Linear layer -> Weight decomposition -> Weight quantization -> Fine-tuned linear layer -> Output

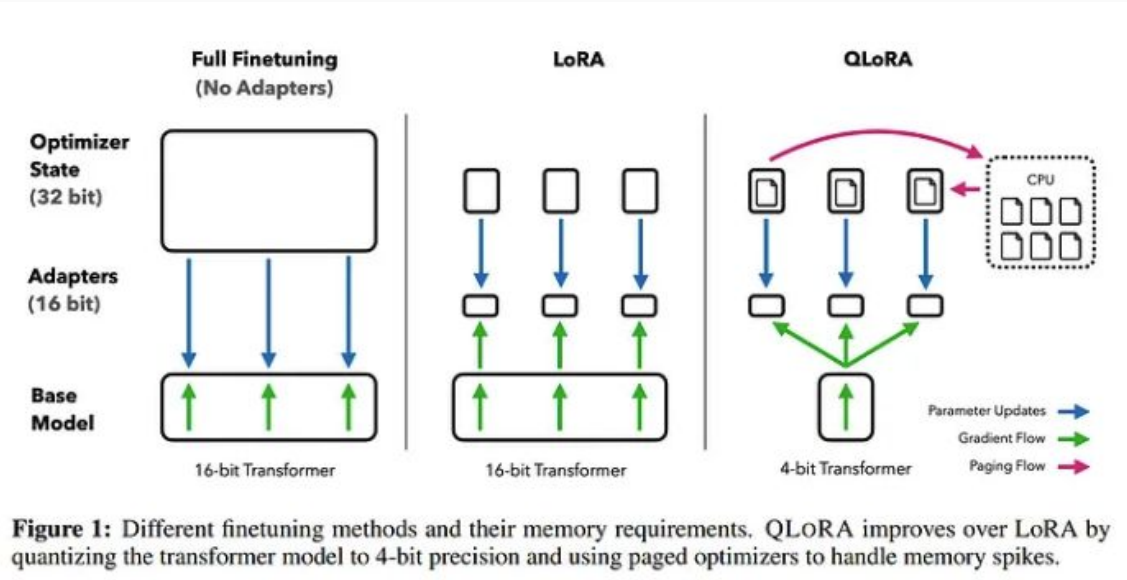


Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

Examination of Building Financial Domain LLM Application - Fine Tuning

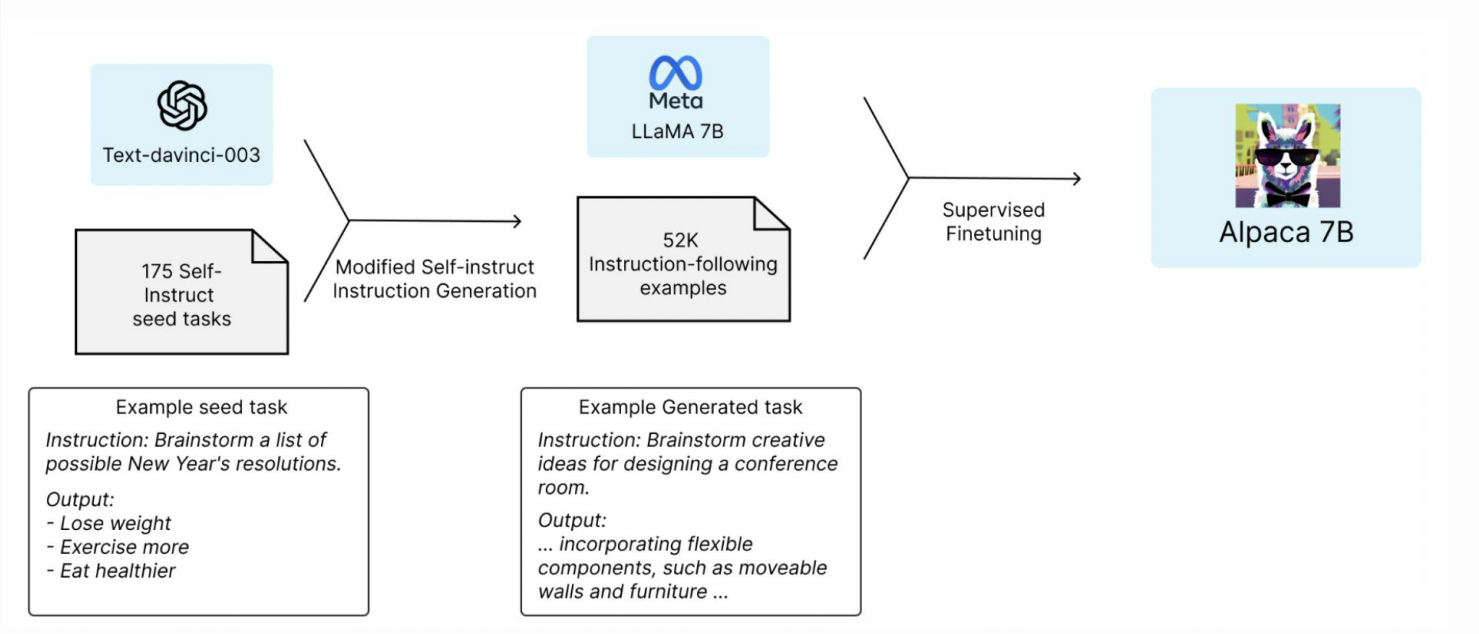
- **Comparison**

LoRA and QLoRA have been shown to achieve similar performance to full fine-tuning on a variety of downstream tasks on the GLUE benchmark

Feature	Fine Tuning	Fine-Tuning with LoRA	Fine Tuning QLoRA
Weight	2 bytes	16 bits	4 bits
Weight gradient	2 bytes	~0.4 bit	~0.4 bit
Optimizer state	8 bytes	~0.8 bit	~0.8 bit
Adapter weights	N/A	~0.4 bit	~0.4 bit
Total bits per parameter	12 bytes	17.6 bits	5.2 bits
Total GPU memory for 70B model	840 GB	154 GB	46 GB
Equivalent number of data center GPUs	18	4	1
Equivalent number of consumer GPUs (RTX 3090 / RTX 4090)	N/A	8	2

Examination of Building Financial Domain LLM Application - Fine Tuning with Distillation

- Examples - Low Cost



Examination of Building Financial Domain LLM - RAG

- **Retrieval augmented generation (RAG)**
  - RAG is a framework for improving the quality of LLM-generated responses by grounding the model on external sources of knowledge.
- **Why RAG?**

  One of major challenge of LLM IS hallucinations. To reduce hallucinations with LLMs is by retrieving useful, factual information and injecting it into the LLM’s prompt as added context.

  RAG helps to address these shortcomings by providing the LLM with access to up-to-date and reliable information.

  RAG also allows users to have insight into the LLM's generative process, by providing the sources that the LLM used to generate its response.
- **Benefits of RAG**

  Improved accuracy and consistency of LLM-generated responses.

  Increased transparency and accountability of LLMs.

  Ability to generate responses that are grounded in external knowledge.

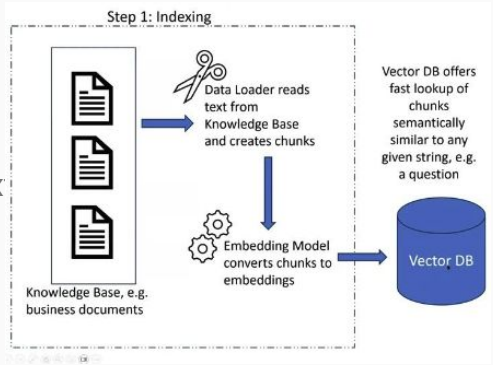
Examination of Building Financial Domain LLM - RAG

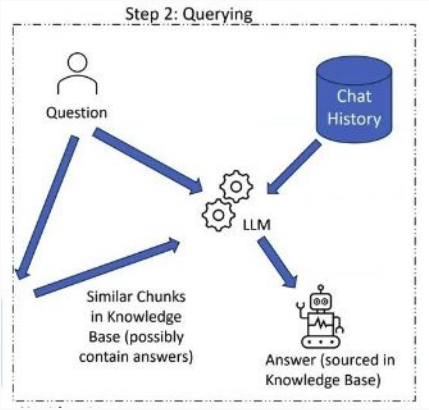
- ## Retrieval augmented generation (RAG) Work flows

1 Chunk the data: Divide the data into chunks of ~200 tokens.

2 Vectorize the chunks: Use an embedding model to vectorize each chunk.

3 Store the vectors: Store the vectors, along with their tex data, in a vector database.(Redis, Weaviate, Pinecone)





4 Retrieve relevant context: At inference time, create a query embedding based on the prompt and run a vector search for relevant documents.

5 Add relevant context to the prompt: Add the relevant documents to the prompt to provide more context to the LLM.

Examination of Building Financial Domain LLM - RAG

- **Query embedding**

  . Simplest approach: Truncate the chat history or prompt and pass it directly into the embedding model.

  If the chat history or prompt is too long, the LLM can be asked to summarize it or convert it into a list of search keywords before embedding it.

- **Embedding model**

  Use a good embedding model that captures the semantic similarities between queries and textual chunks.

  There are a variety of good embedding models publicly available via SentenceTransformers and HuggingFace.

  To find one that works for you, take a look at the Massive Text Embedding Benchmark

- **Benefits**

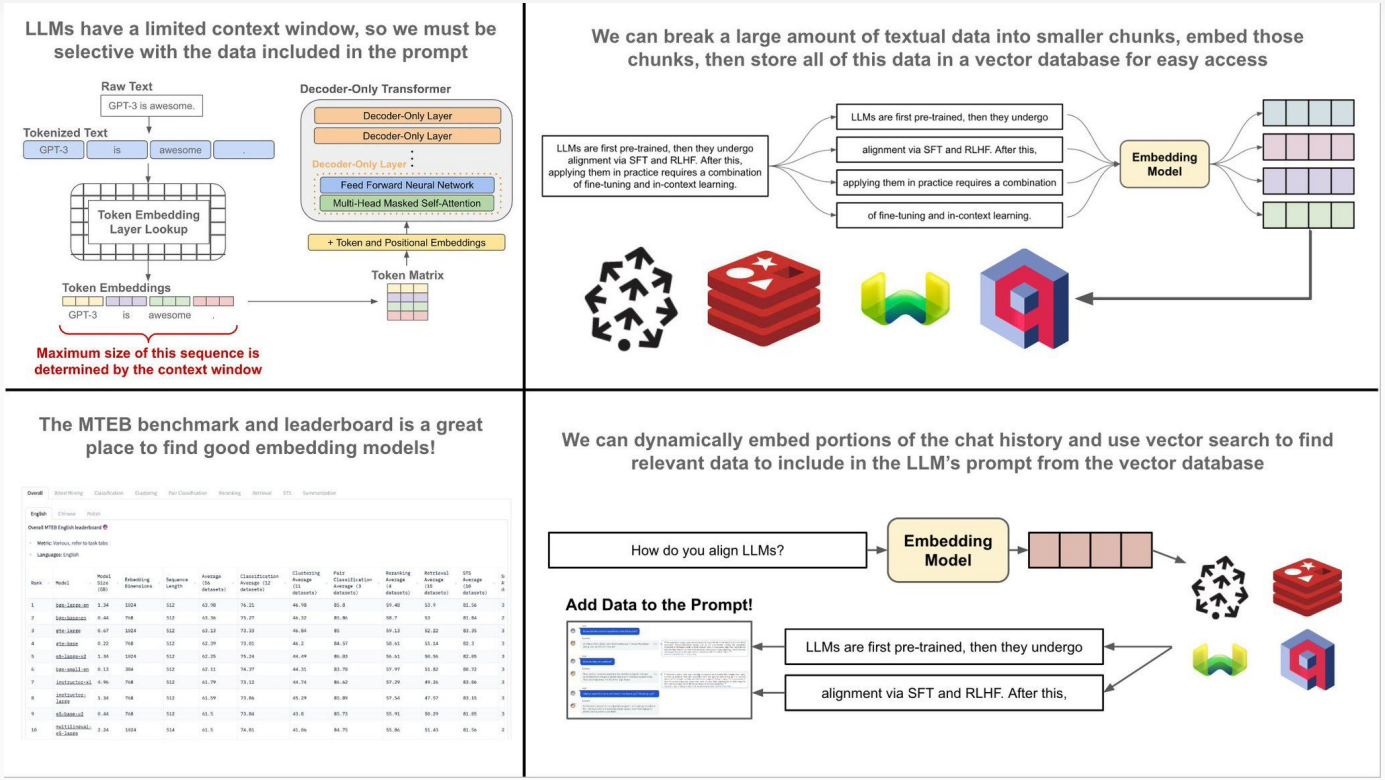
  Improves the accuracy and consistency of LLM-generated responses.

  Allows LLMs to generate responses that are grounded in external knowledge.

  Can be used to scale LLMs to larger datasets.

Examination of Building Financial Domain LLM - RAG

- Retrieval augmented generation (RAG) in details



Examination of Building Financial Domain LLM - Summary

Method	Description	Advantages	Disadvantages	Data Privacy
Fine-tuning	Fine-tuning is a method for improving the performance of a pre-trained LLM on a specific task by training the model on a dataset of examples for that task.	Can achieve the best performance on downstream tasks.	Requires a large dataset of labeled examples for the downstream task. Can be computationally expensive.	Requires access to the downstream task dataset, which may contain sensitive information. Access control should be implemented to ensure that only authorized users have access to the dataset.
Base LLM + few-shot prompting	Few-shot prompting is a method for fine-tuning a pre-trained LLM on a specific task by providing the model with a few examples of the desired output.	Requires much less data than fine-tuning. Can be more efficient in terms of training time and computational resources.	May not achieve the same level of performance as fine-tuning on complex tasks.	Sensitive information may be in the few-shot prompting examples. Access control should be implemented to ensure that only authorized users can provide input to the model.
Base LLM + RAG	Retrieval augmented generation (RAG) is a framework that combines a pre-trained LLM with a retrieval component to improve the accuracy and consistency of the LLM's responses.	Can improve the accuracy and consistency of the LLM's responses, especially on factual tasks. Can be used to generate responses that are grounded in external knowledge.	Requires access to an external knowledge base, which may contain sensitive information. Access control should be implemented to ensure that only authorized users have access to the knowledge base.	Access control should be implemented to ensure that only authorized users can access the retrieved documents.

Examination of Building Financial Domain LLM - Cost Comparison

- **Cost Comparison - Training from Scratch vs Fine-Tuning vs. RAG**

  Feasibility of running a 7-billion-parameter model on a MacBook

  Memory requirements for different precision modes:

  bfloat16: 14 GB memory

  int8: 7GB memory

- **Cost Breakdown for a 7B Parameter Model**

  Fine-Tuning: Approximate cost: $2,000

  Fine-Tuning with Distillation: Approximate cost: $700

  Training from Scratch: Approximate cost: $25,000

  RAG : Approximate cost: $5,000

RAG is a cost-effective way to improve the performance of LLMs on tasks that require access to external knowledge. It is also a good option for users who do not have the resources to fine-tune or train a LLM from scratch.

Operationalizing LLMs for Financial Institution Use-Cases

- **Common Use Cases**

  . Information Extraction, Enterprise Search, Chat with Internal Data, Summarization, Sentiment and Topic, Robo Adviser, Customer Support and QA

- **Examples - Steps to Operationalize LLMs**

  Vectorize the Knowledge: Translate every piece of information in your custom knowledge base into a vector, a multi-dimensional mathematical representation of that information.

  Store and Search the Vectors: Use a vector database to store and look up the vectors efficiently. This will allow you to find similar items based on a query vector, enabling searches for "like" or "close to" other words or phrases, at scale.

  Integration with LLMs: LLMs like GPT-4 do not have information from your custom knowledgebase. When a user sends a prompt/query, you must first look up the vector database for the relevant piece of information.

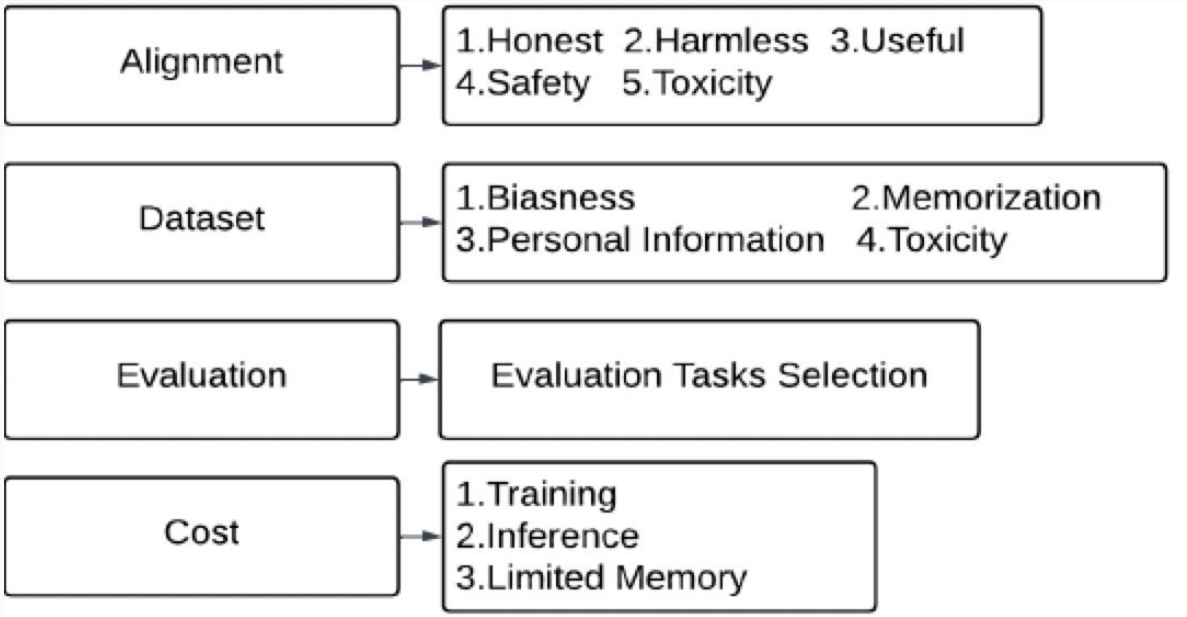
  Orchestration and Operationalization: Developers must come up with a complex orchestration flow to keep state across chats, do multiple vector database look-ups, and facilitate a smooth conversation.

- **Challenges**

  Operationalizing LLMs for enterprise use-cases can be challenging, but it can also offer significant benefits, such as improved search capabilities, more engaging chatbots, and better summarization and key entity extraction.

Examination of Challenges of LLMs

- Four Parts of Challenges when we apply LLM into Finance Industry



Examination of Challenges of LLMs

- **Reduce and measure hallucinations**

  LLMs can sometimes generate text that is factually incorrect or misleading. This is known as hallucination. It is important to be able to reduce and measure hallucination in order to build more reliable LLMs.

- **Optimize context length and context construction**

  LLMs need to be able to understand the context of a query in order to generate relevant and accurate responses. However, it can be difficult to optimize the length and construction of the context, especially for long and complex queries.

- **Incorporate other data modalities**

  LLMs are typically trained on text data, but they can also be trained on other modalities, such as images, audio, and video. This can improve the performance of LLMs on tasks such as image captioning, machine translation, and question answering.

- **Make LLMs faster and cheaper**

  LLMs can be computationally expensive to train and deploy. It is important to make LLMs faster and cheaper in order to make them more accessible to a wider range of users.

- **Build LLMs for non-English languages**

LLMs have the potential to revolutionize many industries and applications. However, there are still a number of challenges that need to be addressed before LLMs can be widely adopted.

Examination of Challenges of LLMs

- **Hallucination in LLM**

  Hallucination occurs when AI models generate information that is not accurate or factual. A major concern for companies considering the adoption of Large Language Models (LLMs).

- **Two Hypotheses on Hallucination**
  - **Understanding of Cause and Effect**
    - Hallucination arises due to LLMs lacking an understanding of cause and effect.
    - Addressed by treating response generation as causal interventions.
  - **Mismatch in Internal Knowledge**
    - Hallucination results from a mismatch between the LLM's internal knowledge and the labeler's knowledge.Behavior cloning during SFT (Supervised Fine-Tuning) can cause hallucination.
- **Proposed Solutions**
  - Verification - Ask the LLM to explain the sources of its answers.
  - Reinforcement Learning (RL) - Improve reward functions to penalize LLMs more for generating false information.

Thanks!