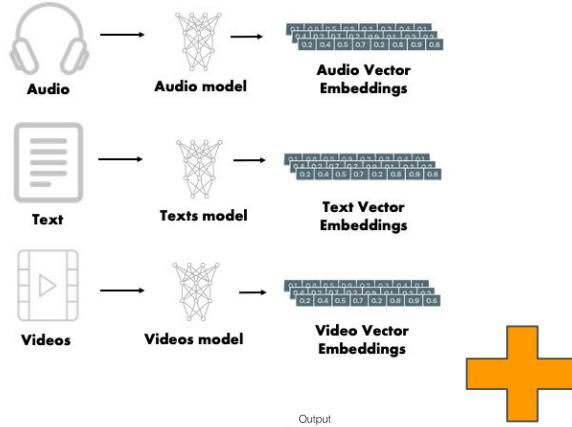


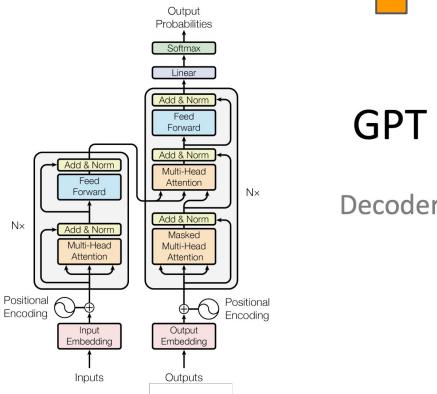
Agentic AI

What are Large Language Model?



BERT

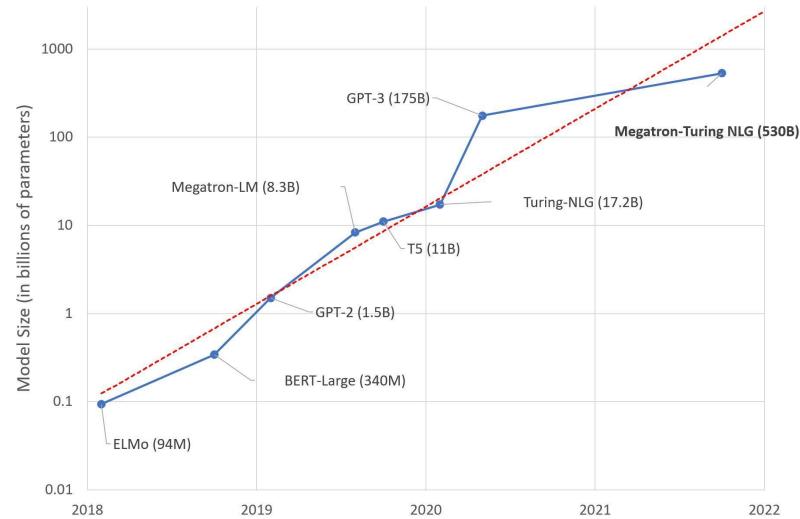
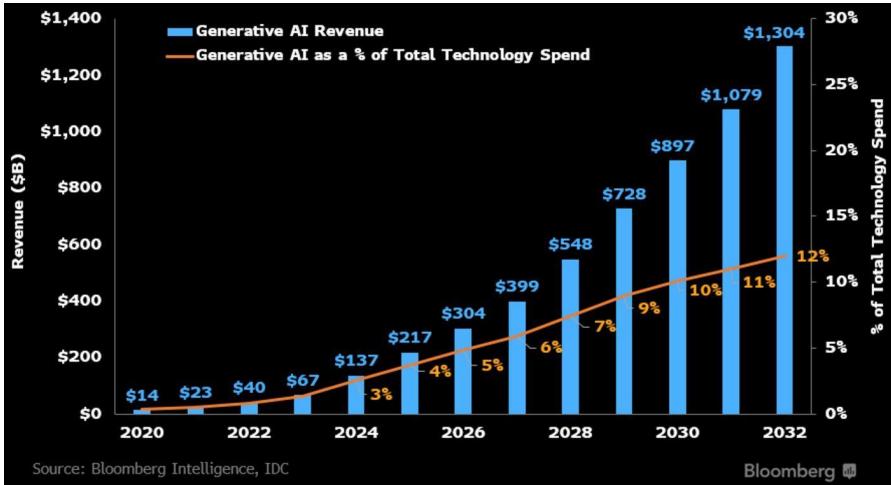
Encoder



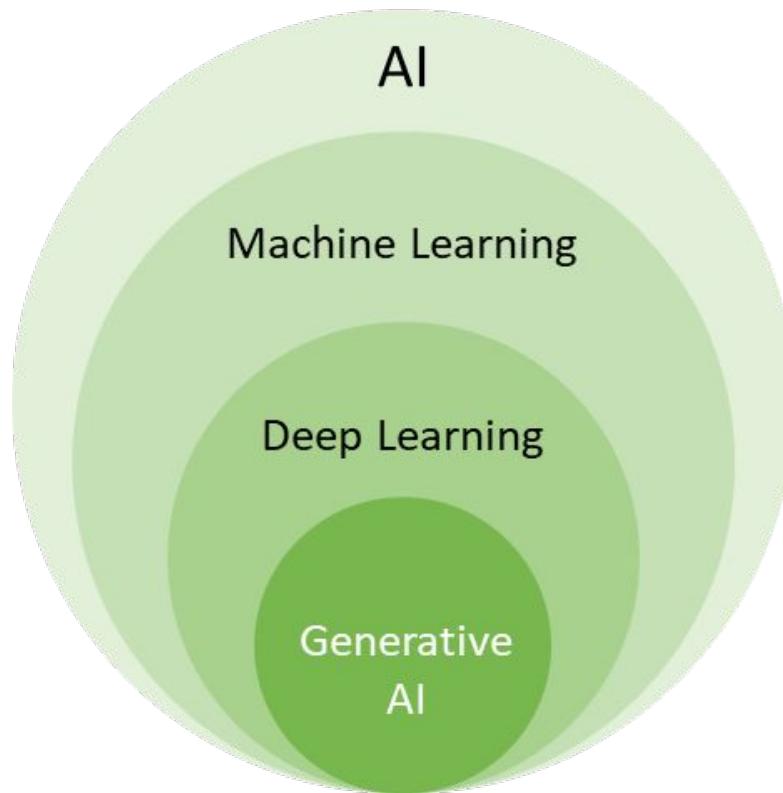
Generative
AI



Generative AI Everywhere!

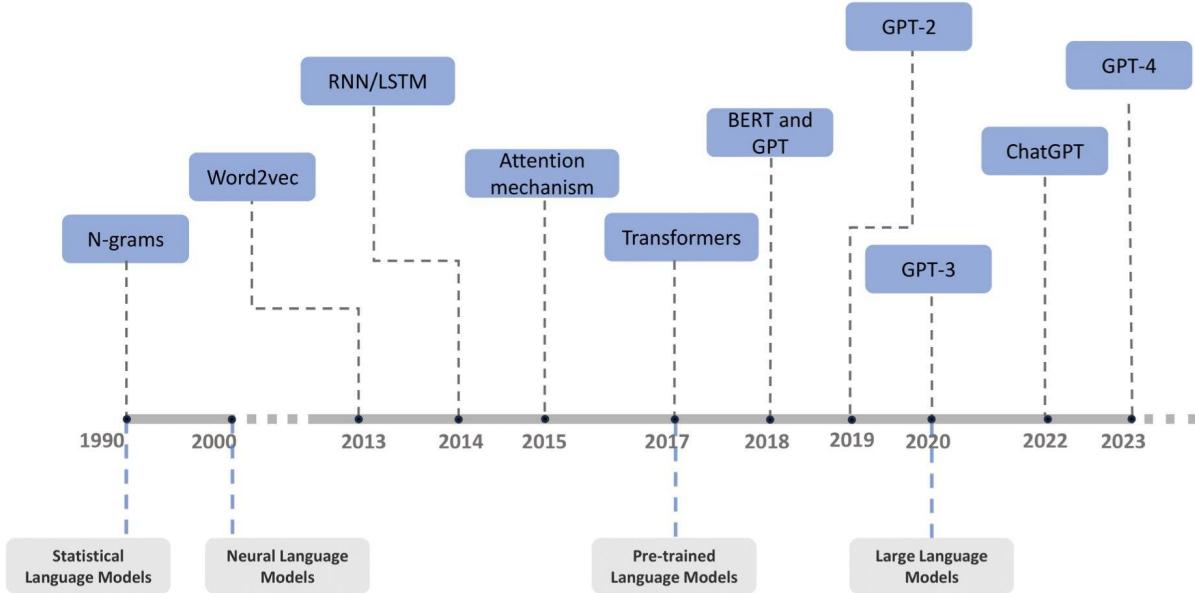


Deep Learning and AI Revolution



History of LLMs

This section is grounded in our introduction to LLMs survey [1].



[1] Wang, Zichong, Chu, Zhibo, Doan, Thang Viet, Ni, Shiwen, Yang, Min, Zhang, Wenbin. "History, development, and principles of large language models: an introductory survey." *AI and Ethics*(2024): 1-17.

History of LLMs

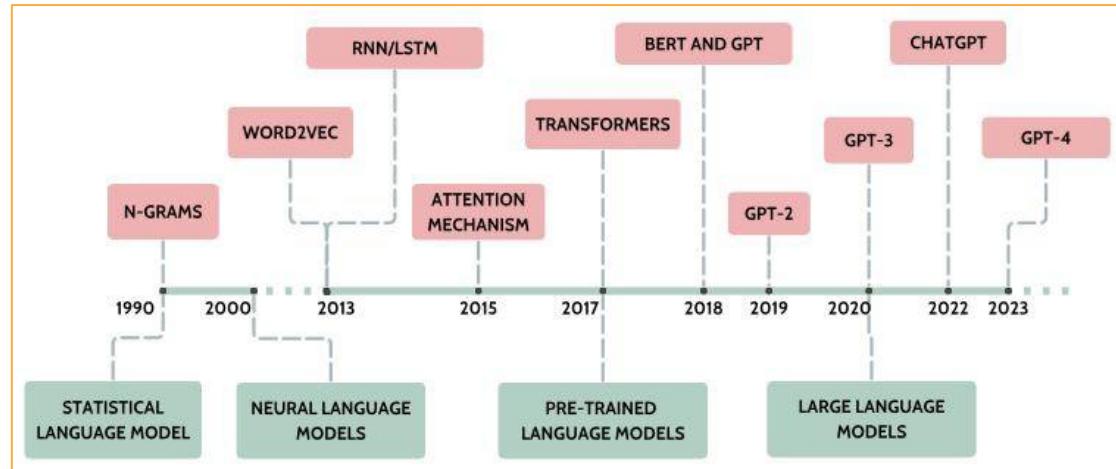
Language Models

- Earlier Stages:
Statistical LMs -> Neural LMs
- N-grams [2]:

$$P(w_n | w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})}$$

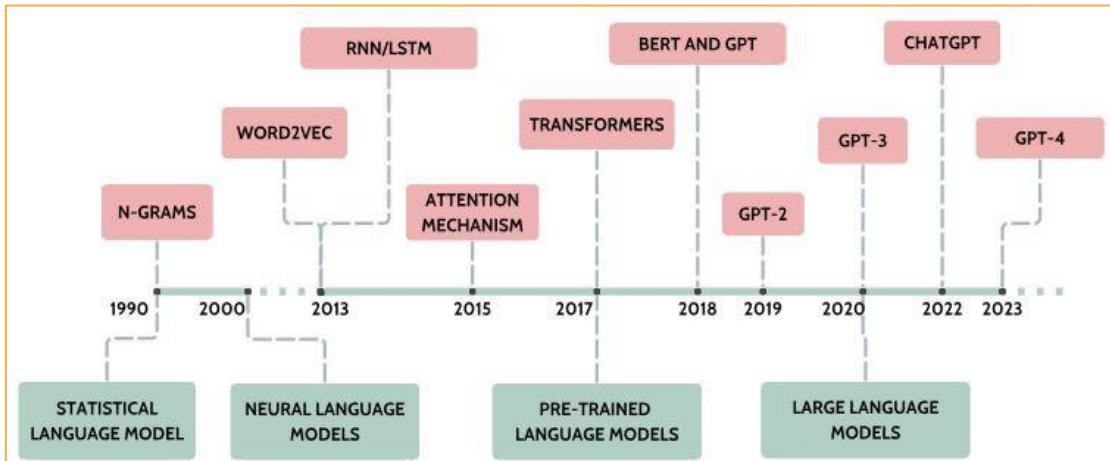
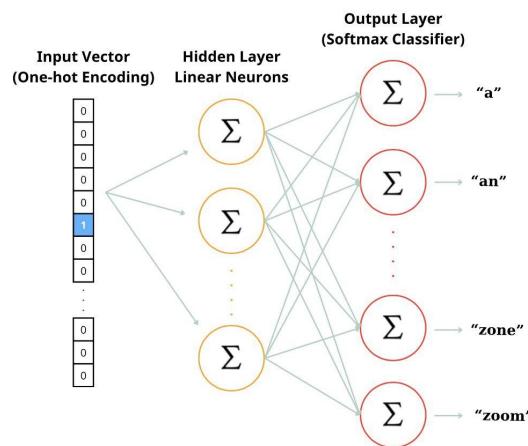
- For example:

He was an engineer in 2002
Bigram : $P(\text{engineer} \text{an}) = \frac{C(\text{an engineer})}{C(\text{an})}$
Trigram : $P(\text{engineer} \text{was an}) = \frac{C(\text{was an engineer})}{C(\text{was an})}$



History of LLMs

Language Models - Statistical/Neural

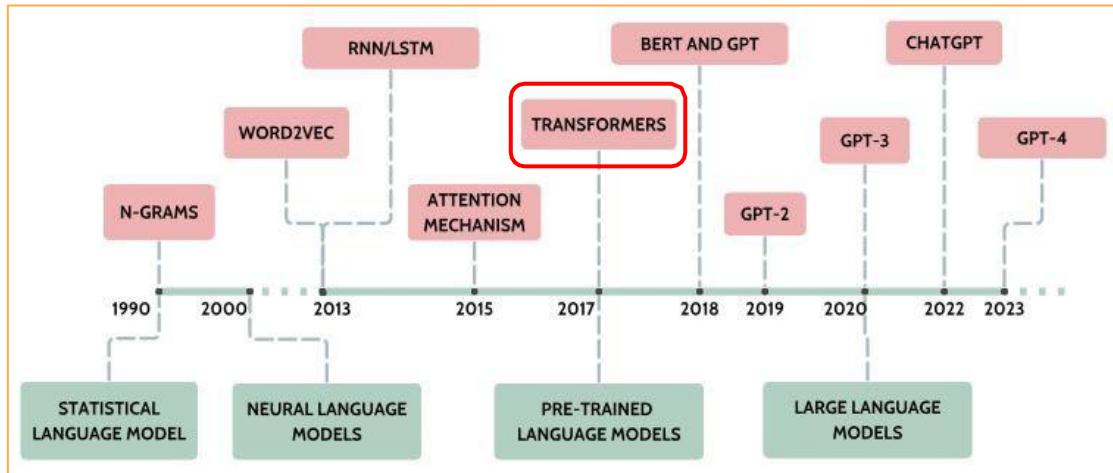


History of LLMs

Large Language Models - Attention

John
read
the
letters
and
they
are
wonderful

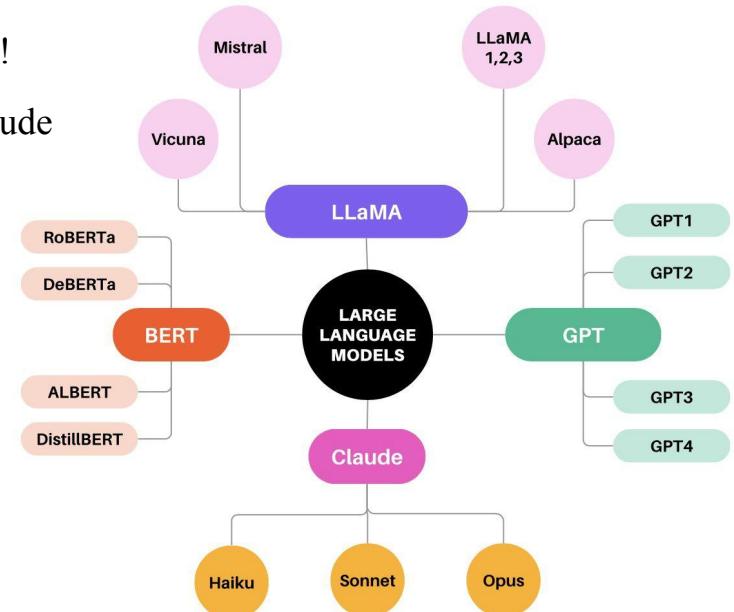
John
read
the
letters
and
they
are
wonderful



History of LLMs

Large Language Models

- Transformers revolutionized the natural language processing landscape!
- Results in a massive blooming era of LLMs: GPT, BERT, LLaMA, Claude and more to go!
- Broad applications across domains:
 - Education
 - Healthcare
 - Technology
 - And so on...



LLMs are not hype—they change the AI game

Generative AI & LLMs are a once-in-a-generation shift in technology

“Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality”



03/30/2023

“Smaller, more performant models such as LLaMA enable... further democratizing access in this important, fast-changing field...”

 Meta AI

02/24/2023

“GPT-4 beats 90% of lawyers trying to pass the bar”

Forbes

03/14/2023

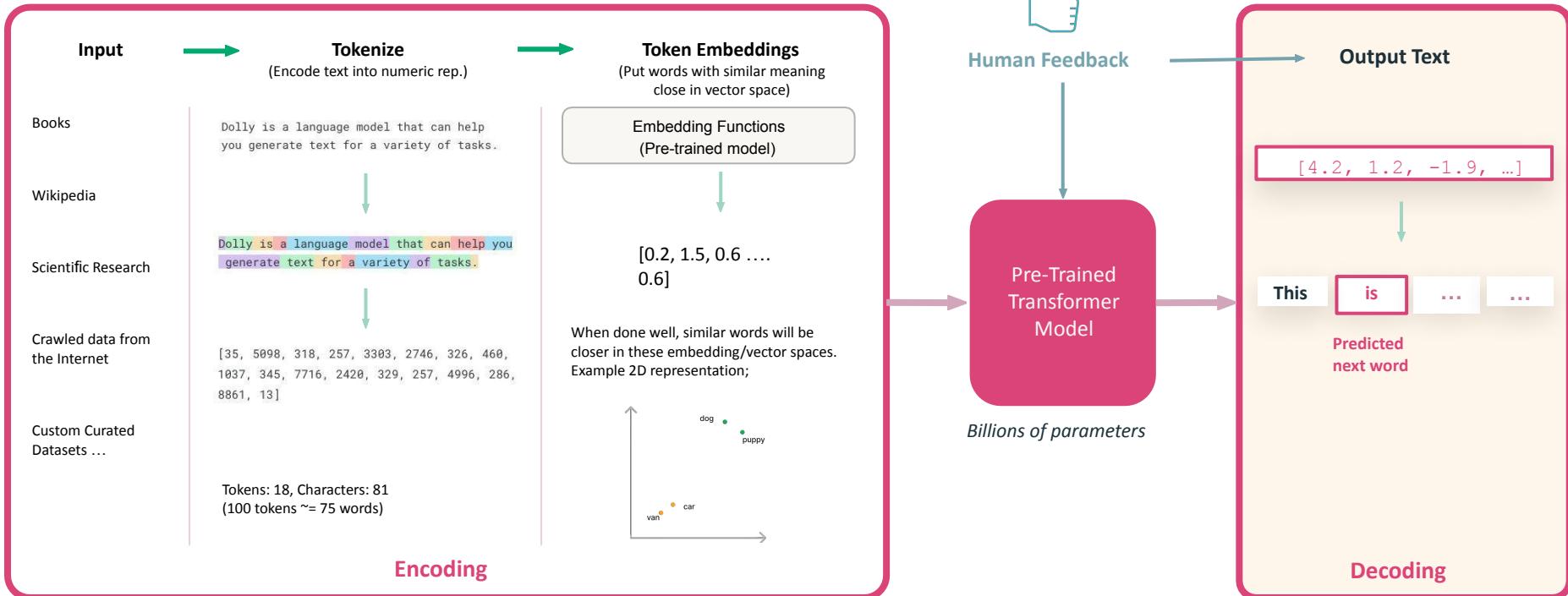
“Falcon is now free of royalties for commercial and research use... Falcon 40B outperforms ... Meta’s LLaMA and Stability AI’s StableLM”

 TII
Technology
Innovation
Institute

05/31/2023

How Do LLMs Work?

A simplified version of LLM training process



courtesy : Databricks



An Overview of Common LLMs

Open-source and Closed LLMs

Model or model family	Model size (# params)	License	Created by	Released	Notes
Falcon	7 B - 40 B	Apache 2.0	Technology Innovation Institute	2023	A newer potentially state-of-the-art model
MPT	7 B	Apache 2.0	MosaicML	2023	Comes with various models for chat, writing etc.
Dolly	12 B	MIT	Databricks	2023	Instruction-tuned Pythia model
Pythia	19 M - 12 B	Apache 2.0	EleutherAI	2023	Series of 8 models for comparisons across sizes
GPT-3.5/4.	175 B	proprietary	OpenAI	2022	ChatGPT model option; related models GPT-1/2/3/4
BLOOM	560 M - 176 B	RAIL v1.0	BigScience	2022	46 languages
FLAN-T5	80 M - 540 B	Apache 2.0	Google	2021	methods to improve training for existing architectures
BART	139 M - 406 M	Apache 2.0	Meta	2019	derived from BERT, GPT, others
BERT	109 M - 335 M	Apache 2.0	Google	2018	early breakthrough



LLMs Generate Outputs for NLP Tasks

Common LLM tasks

	Content Creation and Augmentation	Generating coherent and contextually relevant text. LLMs excel at tasks like text completion, creative writing, story generation, and dialogue generation.
	Summarization	Summarizing long documents or articles into concise summaries. LLMs provide an efficient way to extract key information from large volumes of text.
	Question Answering	Comprehend questions and provide relevant answers by extracting information from their pre-trained knowledge.
	Machine Translation	Automatically converting a text from one language to another. LLMs are also capable to explain language structure such as grammatical rules.
	Classification	Categorizing text into predefined classes or topics. LLMs are useful for tasks like topic classification, spam detection, or sentiment analysis.
	Named Entity Recognition (NER)	Identifying and extracting named entities like names of persons, organizations, locations, dates, and more from text.
	Tone / Level of content	Adjusting the text's tone (professional, humorous, etc.) or complexity level (e.g., fourth-grade level).
	Code generation	Generating code in a specified programming language or converting code from one language to another.



LLMs Business Use Cases

Customer Engagement

- Personalization and customer segmentation:
 - Provide personalized product/content recommendation based on customer behaviour and preferences
- Feedback Analysis
- Virtual assistants

What are the top 5 customer complaints based on the provided data?



Based on the **customer review dataset**, the top 5 customer complaints are as follows:

1. **Shipping Delays** - 25% of customers expressed frustration with delayed deliveries.
2. **Product Quality** - 20% of customers reported issues with the quality of the received products.
3. **Customer Service Responsiveness** - 18% of customers felt that the response time from customer service was slow.
4. **Billing and Payment Errors** - 15% of customers encountered errors and discrepancies in their bills and payments.
5. **Order Inaccuracies** - 12% of customers received incorrect or incomplete orders.

Databricks

LLMs Business Use Cases

Content Creation

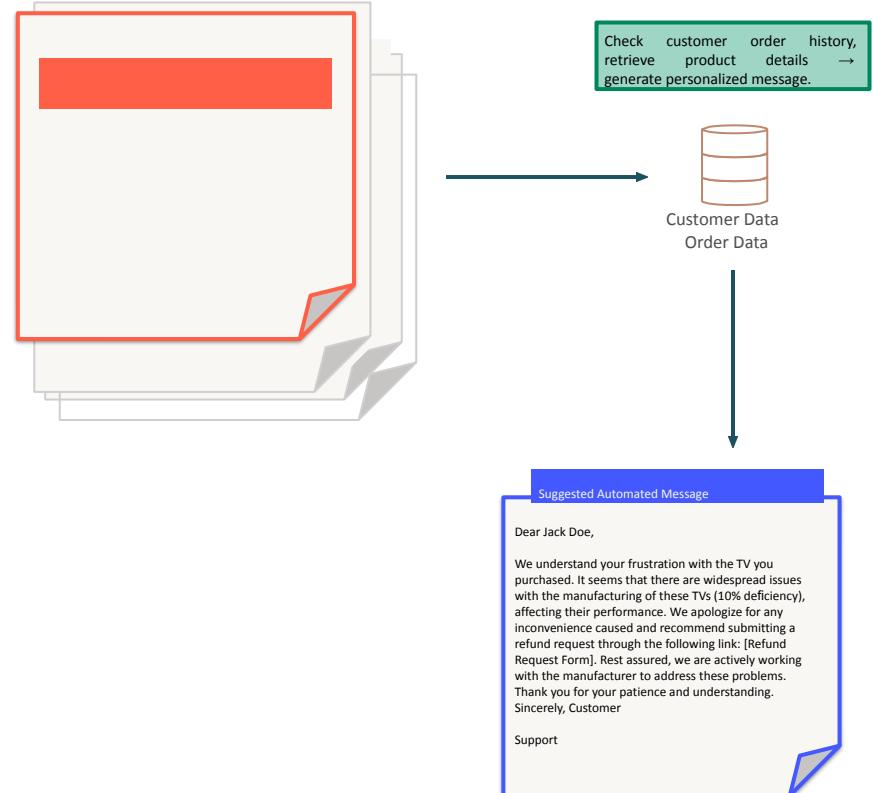
- Creative writing: Short stories, creative narratives, scripts etc.
- Technical writing: Documentation, user manuals, simplifying content etc.
- Translation and localization
- Article writing for blogs/social media



LLMs Business Use Cases

Process automation and efficiency

- Customer support augmentation and automated question answering
- Automated customer response
 - Email
 - Social media, product reviews
- Sentiment analysis, prioritization



courtesy : Databricks



LLMs Business Use Cases

Code generation and developer productivity

- Code completion, boilerplate code generation
- Error detection and debugging
- Convert code between languages
- Write code documentation
- Automated testing
- Natural language to code generation
- Virtual code assistant for learning to code

The image shows a screenshot of the Copilot interface, which is a developer productivity tool. It displays two code snippets side-by-side.

sentiments.ts:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch('http://text-processing.com/api/sentiment/', {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

max_sum_slice.py:

```
1 def max_sum_slice(xs):
2     if not xs:
3         return 0
4
5     max_ending = max_slice = 0
6     for x in xs:
7         max_ending = max(0, max_ending + x)
8         max_slice = max(max_slice, max_ending)
9     return max_slice
```



LLM Flavors

Thinking of building your own modern LLM application?



Open-Source Models

- Use as **off-the-shelf** or **fine-tune**
- Provides flexibility for customizations
- Can be smaller in size to save cost
- **Commercial / Non-commercial use**

Open-source LLMs:

Non-commercial Use

Commercial Use

Meta AI
LlaMA

databricks
Dolly

mosaic^{ML}
MPT



Proprietary Models

- Usually offered as **LLMs-as-a-service**
- Some can be **fine-tuned**
- Restrictive licenses for usage and modification

Proprietary LLMs:

ANTHROPIC



OpenAI



PaLM 2

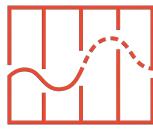
Choose the right LLM model flavor

There is no “perfect” model, trade-offs are required.

LLM model decision criteria



Privacy



Quality



Cost



Latency

Using Proprietary Models (LLMs-as-a-Service)

Pros

- Speed of development
 - Quick to get started and working.
 - As this is another API call, it will fit very easily into existing pipelines.
- Quality
 - Can offer state-of-the-art results

Cons

- Cost
 - Pay for each token sent/received.
- Data Privacy/Security
 - You may not know how your data is being used.
- Vendor lock-in
 - Susceptible to vendor outages, deprecated features, etc.

Using Open Source Models

Pros

- Task-tailoring
 - Select and/or fine-tune a task-specific model for your use case.
- Inference Cost
 - More tailored models often smaller, making them faster at inference time.
- Control
 - All of the data and model information stays entirely within your locus of control.

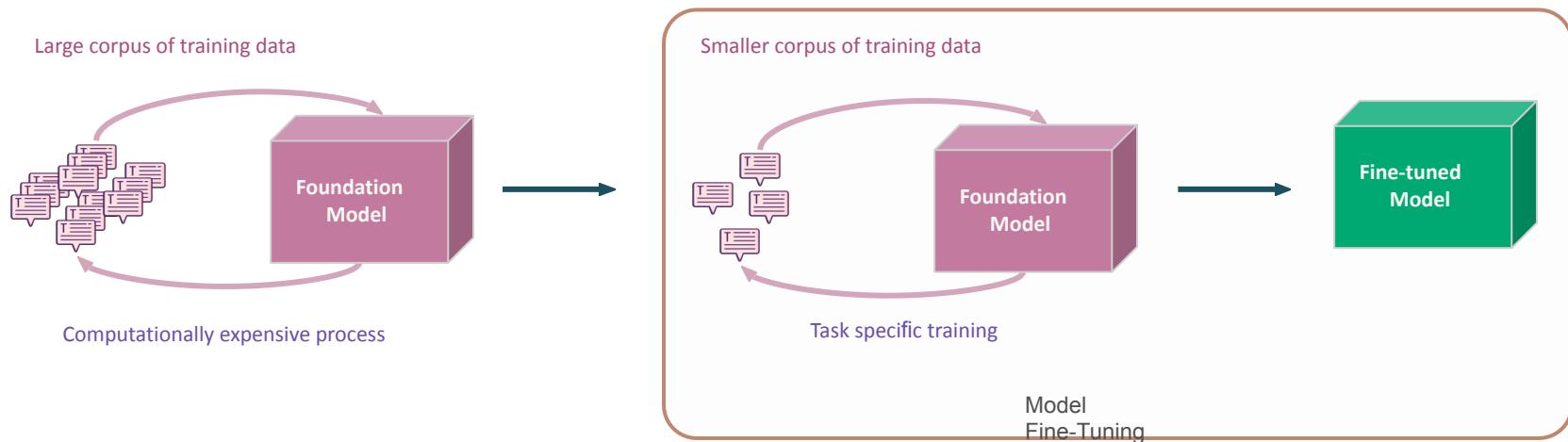
Cons

- Upfront time investments
 - Needs time to select, evaluate, and possibly tune
- Data Requirements
 - Fine-tuning or larger models require larger datasets.
- Skill Sets
 - Require in-house expertise

Fine Tuned Models

What is fine-tuning and how it works

Fine-tuning: The process of further training a pre-trained model on a specific task or dataset to adapt it for a particular application or domain.

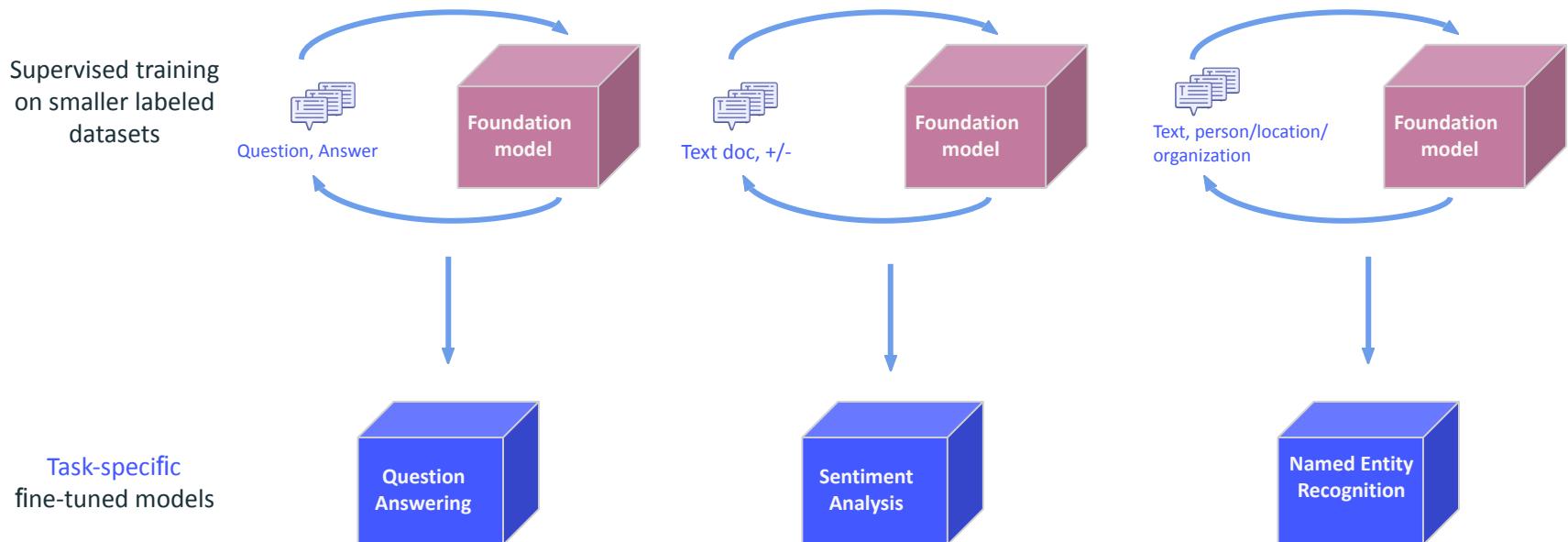


courtesy : Databricks



Fine-tuning models

Foundation models can be fine-tuned for **specific tasks**

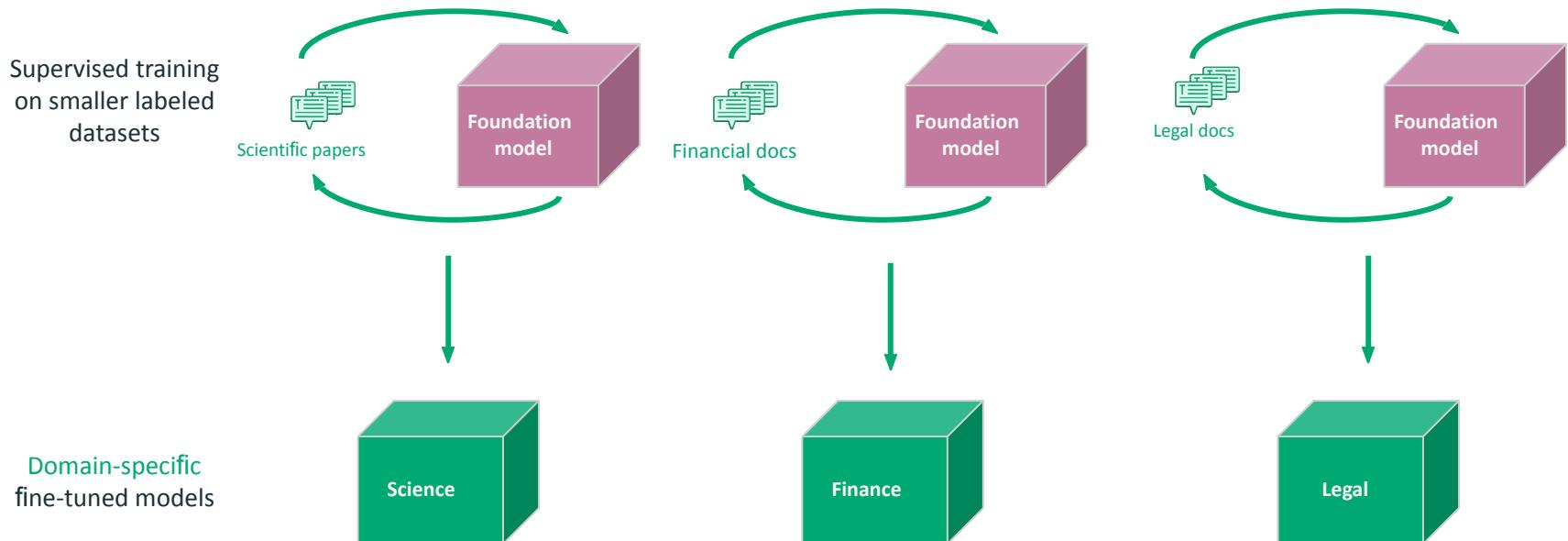


courtesy : Databricks



Fine-tuning models

Foundation models can be fine-tuned for **domain adaptation**

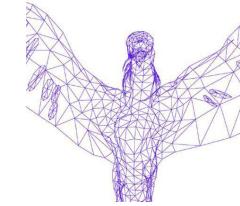
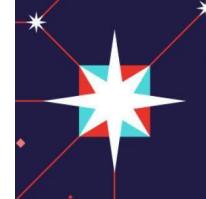


courtesy : Databricks



Open Source quality is rapidly advancing – while fine tuning cost is rapidly decreasing

Dolly started the trend to open models with a commercially friendly license



Facebook LLaMA

Stanford Alpaca

Databricks Dolly Mosaic MPT TII Falcon

"Smaller, more performant models such as LLaMA ... democratizes access in this important, fast-changing field."

"Alpaca behaves qualitatively similarly to OpenAI ... while being surprisingly small and easy /cheap to reproduce"

"Dolly will help democratize LLMs, "MPT-7B is trained from scratch on transforming them into a commodity every company can own and customize"

"Falcon significantly outperforms GPT-3 for ... 75% of the training tokens ... is open source, available for commercial use, and matches the quality of LLaMA-7B"

February 24, 2023

March 13, 2023

March 24, 2023

May 5, 2023

May 24, 2023

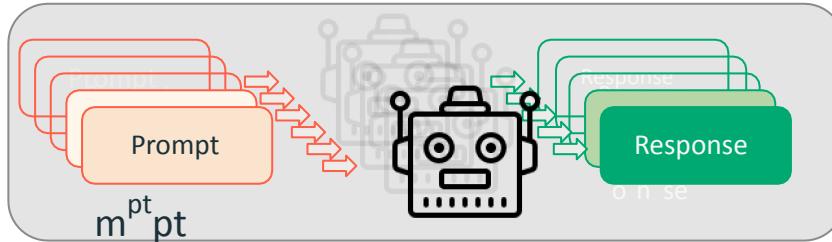
Non Commercial Use Only | **Commercial Use Permitted**



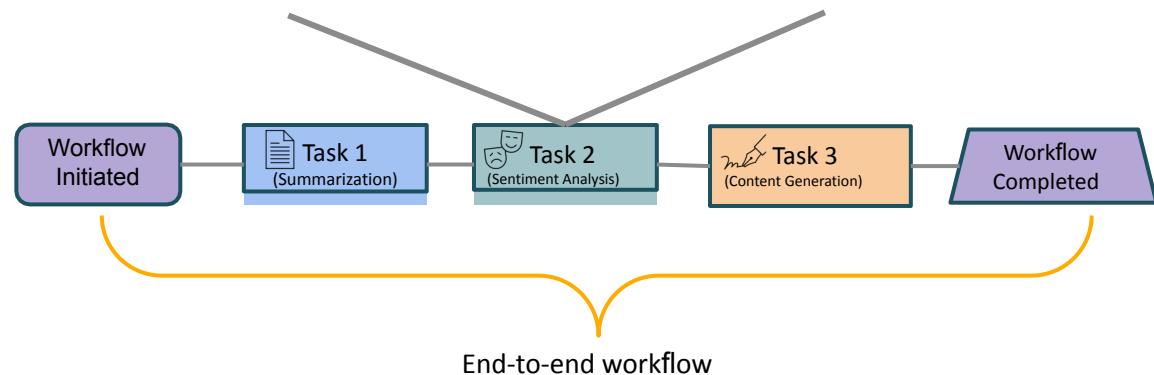
Mixing LLM Flavors in a Workflow

Typical applications are more than just a prompt-response system.

Tasks: Single interaction
with an LLM



Workflow: Applications
with more than a single
interaction



courtesy : Databricks

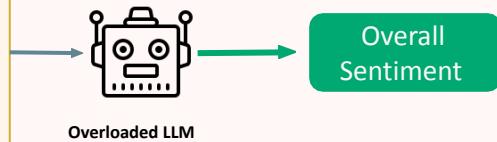


Mixing LLM Flavors in a Workflow

Example multi-LLM problem: get the sentiment of many articles on a topic



Article 1: "...
Article 2: "...
Article 3: "...
Article 4: "...
Article 5: "...
Article 6: "...
Article 7: "...
..."



Initial solution

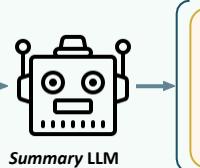
Put all the articles together and have the LLM parse it all

Issue

Can quickly overwhelm the model input length



Article 1:
"...
Article 2:
"
..."
Article 3:
..."



Better solution

A two-stage process to first summarize, then perform sentiment analysis.



Delivering business value from Gen AI is challenging. How do we...?



Customize LLMs with
our data



Ensure LLMs deliver
high quality answers



Securely connect our
data to LLMs



Integrate LLMs with
data governance



Deploy LLMs without
new infrastructure



Maintain flexibility to
upgrade LLMs

Prompt Engineering

Generative AI can create text, images, video, and more. Prompt engineering is the process of refining what you ask a generative AI tool to do. Anyone can do this using natural language in generators like ChatGPT or DALL-E. It's also a technique AI engineers use when refining large language models (LLMs) with specific or recommended prompts.

LLM Prompt Template

INSTRUCTIONS

"""Answer the question based on the context below. If the question cannot be answered using the information provided answer with "I don't know".

CONTEXTS (EXTERNAL INFO)

Context: Large Language Models (LLMs) are the latest models used in NLP. Their superior performance over smaller models has made them incredibly useful for developers building NLP enabled applications. These models can be accessed via Hugging Face's `transformers` library, via OpenAI using the `openai` library, and via Cohere using the `cohere` library.

Question: Which libraries and model providers offer LLMs?

Answer: """

PROMPTER INPUT

OUTPUT INDICATOR

Example Prompting Strategies

SYSTEM

You will be provided with customer service queries. Classify each query into a primary category and a secondary category. Provide your output in json format with the keys: primary and secondary.

Primary categories: Billing, Technical Support, Account Management, or General Inquiry.

Billing secondary categories:

- Unsubscribe or upgrade
- Add a payment method
- Explanation for charge
- Dispute a charge

Technical Support secondary categories:

- Troubleshooting
- Device compatibility
- Software updates

Account Management secondary categories:

- Password reset
- Update personal information
- Close account
- Account security

General Inquiry secondary categories:

- Product information
- Pricing
- Feedback
- Speak to a human

USER

I need to get my internet working again.

[Open in Playground ↗](#)

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27.

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.

Design Patterns with LLMs

Prompt Components

A prompt consists of:

- **Instruction:** a specific task or instruction you want the model to perform
- **Context:** external information or additional context that can steer the model to better responses
- **Input Data:** the input or question that we are interested to find a response for

Zero shot one shot ... few shot?

- LLMs are tuned to follow instructions (instruction tuned or LLMIT)
- Capable of performing some tasks in a "zero-shot" manner.
- The prompt won't contain examples or demonstrations.
- Instructs the model to perform a task without any additional examples to steer it.

“Classify the text into neutral, negative or positive.

Text: I think the vacation is okay.

Sentiment:

Few Shot Prompt

- For complex task, zero-shot may not enough
- Few-shot enables **in-context learning** where we provide demonstrations in the prompt
 - This steers the model to better performance.

⇒

"A "whatpu" is a small, furry animal native to Tanzania.

An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

⇒

To do a "farduddle" means to jump up and down really fast.

An example of a sentence that uses the word farduddle is:

"

Few-shot = In-context Learning

- This is pretty much the same thing

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // _____



Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // _____



Few Shot Prompt

The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.

A: The answer is False.



The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.

A: The answer is True.

The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.

A: The answer is True.

The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.

A: The answer is False.

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.

A:

The answer is True.



Chain-of-Thought (CoT) Prompting

The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.

A: Adding all the odd numbers (9, 15, 1) gives 25. The answer is False.



The odd numbers in this group add up to an even number: 17, 10, 19, 4, 8, 12, 24.

A: Adding all the odd numbers (17, 19) gives 36. The answer is True.

The odd numbers in this group add up to an even number: 16, 11, 14, 4, 8, 13, 24.

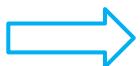
A: Adding all the odd numbers (11, 13) gives 24. The answer is True.

The odd numbers in this group add up to an even number: 17, 9, 10, 12, 13, 4, 2.

A: Adding all the odd numbers (17, 9, 13) gives 39. The answer is False.

The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.

A:



Adding all the odd numbers (15, 5, 13, 7, 1) gives 41. The answer is False.

Chain-of-Thought (CoT) Prompting

The odd numbers in this group add up to an even number: 4, 8, 9, 15, 12, 2, 1.
A: Adding all the odd numbers (9, 15, 1) gives 25. The answer is False.



The odd numbers in this group add up to an even number: 15, 32, 5, 13, 82, 7, 1.
A:

Even fewer examples work!



Adding all the odd numbers (15, 5, 13, 7, 1) gives 41. The answer is False.

Chain-of-Thought (CoT) Prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. X

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Zero-shot COT Prompting

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. X

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 X

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: Let's think step by step.

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

What can you do with a Pre-trained LLM?

Easy / Simple



More Difficult

- Basic chat or using embeddings in downstream tasks
- Prompt Engineering (few shot, CoT, templates, chains, etc.)
- Retrieval Augmented Generation (RAG)
- Supervised Fine-Tuning (SFT)
- Autonomous Agents connected to external tools

Content

- Review the development **history** of LLMs
- **Training procedure** of LLMs, how it achieves such capabilities
- Explore the **bias sources** in LLMs



What challenges does the retrieval augmented generation approach solve?

Problem 1: LLM models do not know *your data*

Problem 2: AI applications must leverage *custom data* to be effective

Use case:

Question and answer chatbots

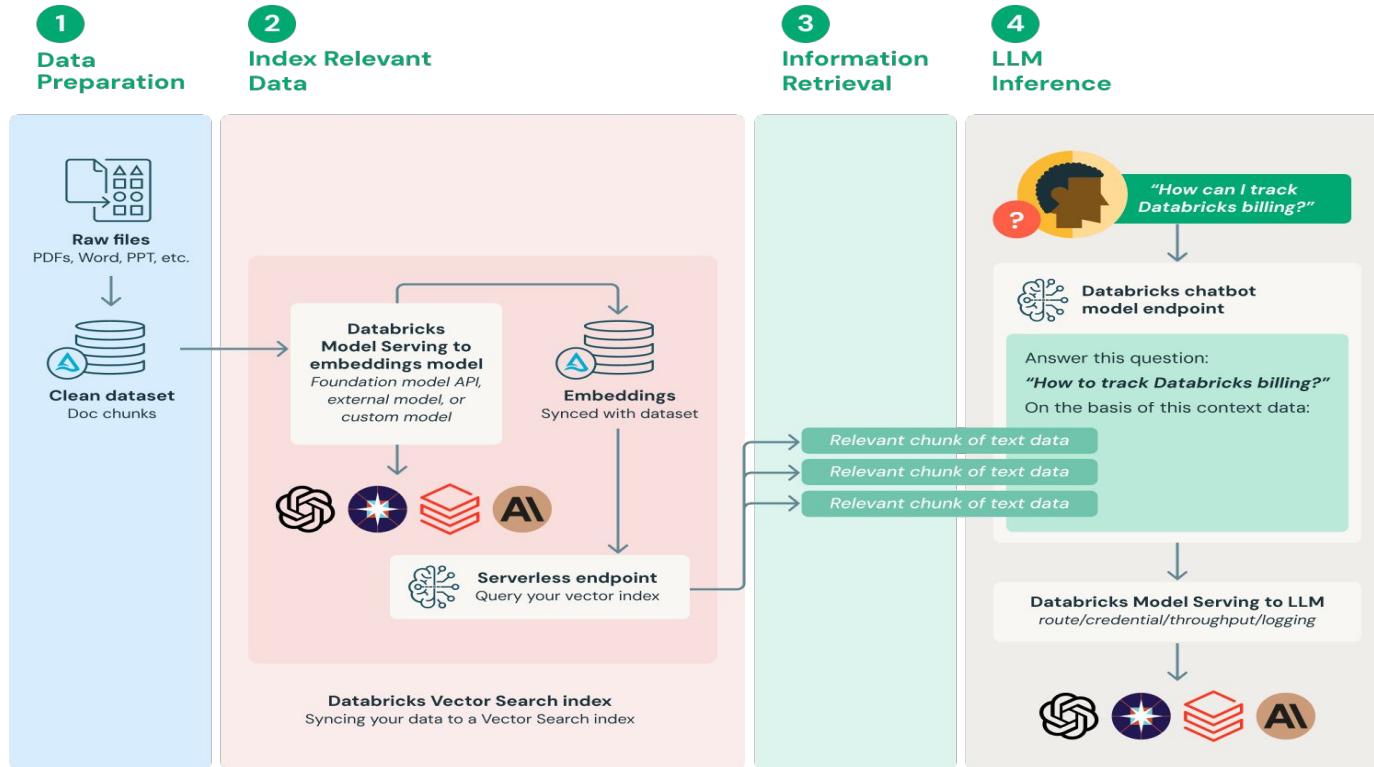
Search augmentation

Knowledge engine — ask questions on your data

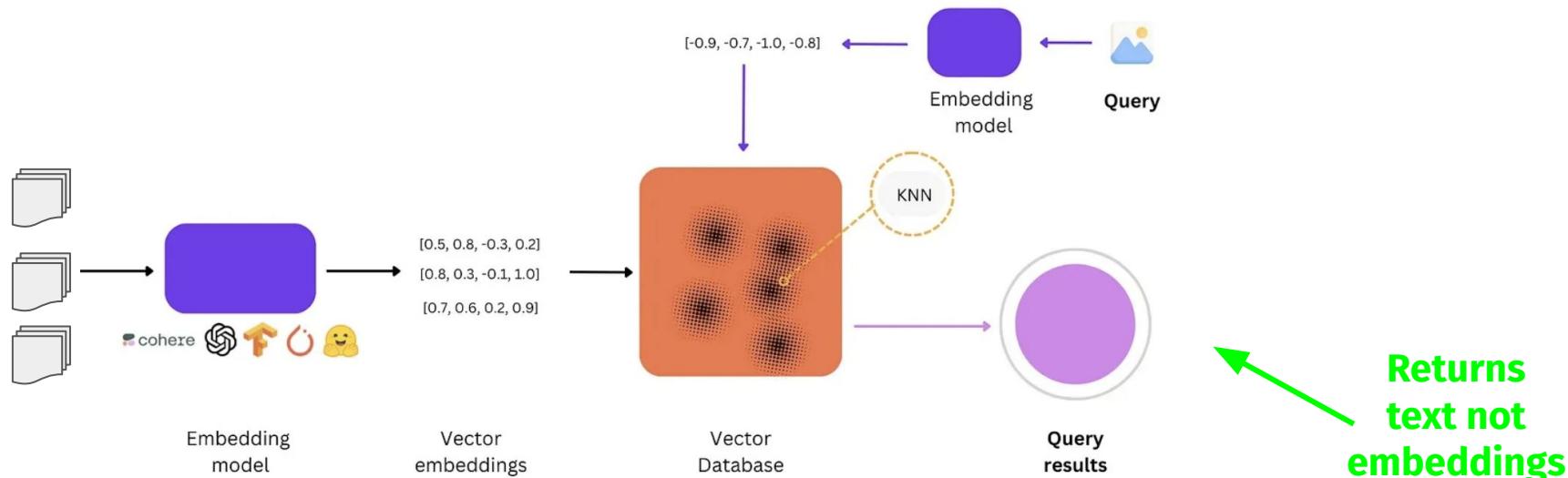
Fine-Tune vs RAG

RAG is the right place to start, being easy and possibly entirely sufficient for some use cases. Fine-tuning is most appropriate in a different situation, when one wants the LLM's behavior to change, or to learn a different "language." These are not mutually exclusive. As a future step, it's possible to consider fine-tuning a model to better understand domain language and the desired output form — and also use RAG to improve the quality and relevance of the response.

RAG Applications



Vector Store



RAG (Indexing)

Indexing is the data preparation step where the data on which retrieval is performed is extracted and cleaned from data sources like files, URLs etc. and converted to plain text. This plain text is commonly a string of multiple thousands of characters. For instance, say you want to perform question answering from your university textbook with help of LLMs. The entire book cannot be fed to the LLM since it might exceed the context window of the LLM. Hence, we break the entire content into smaller and manageable pieces called chunks, and this process is called chunking. These chunks are then transformed into high-dimensional vectors with help of embedding models.

RAG (Retrieval)

This is a crucial step in the process where once the user query is received, the same embedding model used to create vectors of the data sources is used to encode the query into a vector. This vector is used to perform similarity search over the pool of vector embeddings of the data sources and find out the most similar chunks. The similarity search methods that are mostly employed are cosine similarity, dot product, euclidean distance etc. The retrieved chunks are used in the generation step to get response.

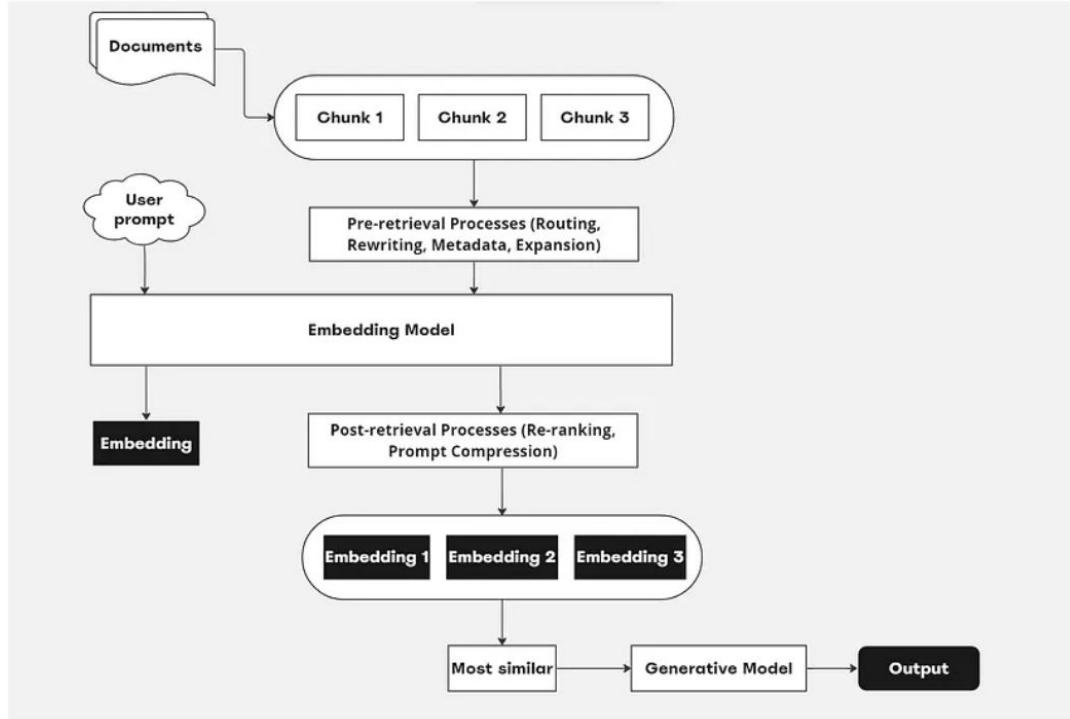
RAG(Generation)

The retrieved chunks along with the user query and instructions are coalesced into a single prompt, which is then fed to the LLM for generating response. Due to the addition of extra information to the prompt, the LLM has sufficient data to generate relevant response. It can be restricted to provide response adhering only to the retrieved chunks. Feeding the chat history of the interactive conversation between the LLM and the user also improves the quality of responses.

Shortcomings - Naive RAG

Naive RAG approach has a lot of shortcomings when it comes to reducing hallucinations. This approach suffers from low precision when relevant chunks are not retrieved properly. The presence of outdated data in the corpus leads to inaccurate chunk retrieval. The response generated may not be grounded on the additional context provided. The sequencing of chunks does matter. The order in which the chunks are arranged should be according to the relevance to the query. There's also a risk of constraining the response and it just containing the retrieved information

Advanced RAG



Advanced RAG

Pre-Retrieval Processes

The goal of this process is to improve the quality of content being indexed. The standard of text from data sources is enhanced by removing irrelevant information, removing ambiguity and inaccuracies, maintaining context, updating outdated documents etc. Adding suitable metadata to each chunk significantly improves the quality of retrieved documents. There might be scenarios where the user query may not be the best prompt to feed to the LLM. So, query rewriting techniques are implemented where we rewrite the query based on the LLM's characteristics to improve the quality of generation.

Post-Retrieval Processes

After the high-quality chunks are retrieved, the way in which these are merged with the user query to form the prompt influences the quality of generation. Simply appending chunks to the user query may exceed the context window limit, introduce noise and degrade the response quality. So, additional techniques like re-ranking, prompt compression are implemented to overcome these challenges.

Reranking technique is implemented to reorder the sequence of retrieved chunks based on the contextual similarity between the chunk and the user query rather than just comparing the vector similarity. Prompt compression technique is used to reduce noise in the retrieved documents. This includes compressing irrelevant information, highlighting important passages, reducing context length etc.

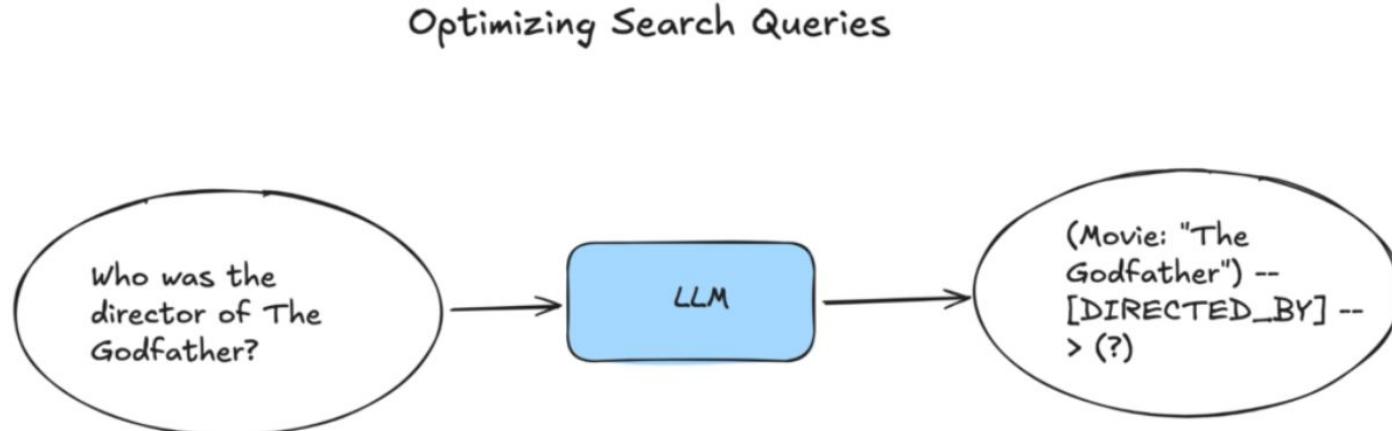
Search Techniques

- **Full text search**
- **Vector search**
- **Hybrid search**
- **Semantic Search**

Retrieval Techniques

Optimize Search Queries Using LLMs

This technique restructures the user's query into a format that is more understandable by the LLM and usable by retrievers. Here, you first process the user query through a fine-tuned language model to optimize and structure it. This process removes any irrelevant context and adds necessary metadata, ensuring the query is tailored to the underlying data store.



Retrieval Techniques

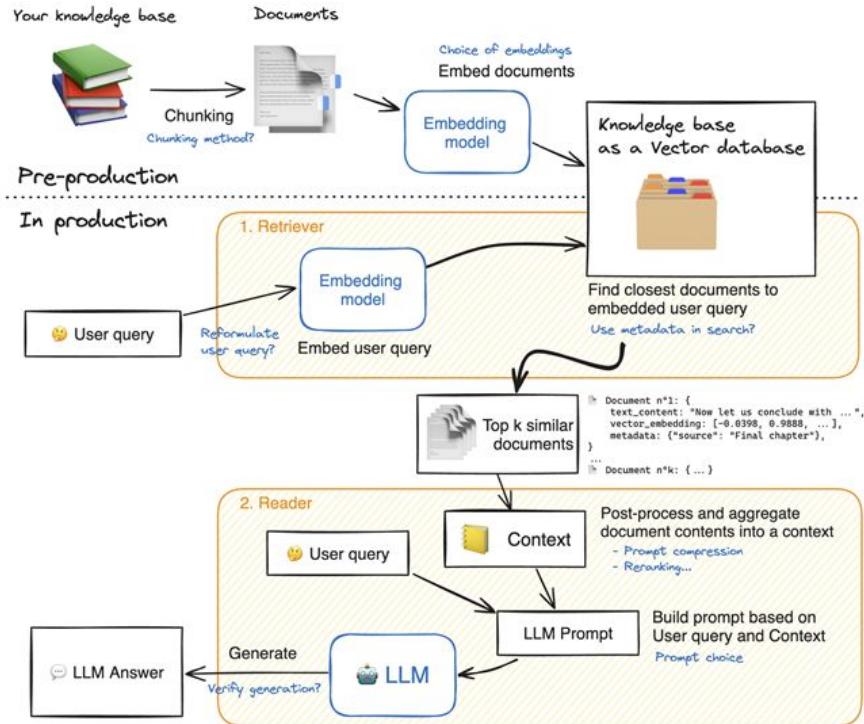
Apply Hierarchical Index Retrieval

You can also use hierarchical indexing to enhance the precision of RAG applications. In this approach, data is organized into a hierarchical structure, with information categorized and sub-categorized based on relevance and relationships.

The retrieval process begins with broader chunks or parent nodes, followed by a more focused search within smaller chunks or child nodes linked to the selected parent nodes. Hierarchical indexing not only improves retrieval efficiency but also minimizes the inclusion of irrelevant data in the final output.

Decision Making in RAG

- Pre-built or custom data loader?
- How to encode documents?
- How to chunk documents?
- How to encode queries?



- How to evaluate and monitor performance?
- How much and what do I retrieve?
- How to prompt and pass in the context?
- Do I need tools / agents?

Cool Demos

Langchain

https://colab.research.google.com/drive/1QA7h2qE4zzSW5GmsSfPDNV1dXDRhAT_H#scrollTo=uIdQk4LZBx6t

Rag

https://colab.research.google.com/drive/1eTE7iWrJd5rHaYR2J_PIH1DgtLhOxfz9#scrollTo=GtXIIPZ3v8Uq

<https://colab.research.google.com/drive/1CqPLPgX0kU177ef8GVqmmVDYHObRiJf1#scrollTo=wUVb-9mKs3AH>

React:

https://colab.research.google.com/drive/1y7vh-kpIT5H7Q9WF90lZFNOu-rzhqUh_m#scrollTo=5EpE774KgOPu

ReAct: Reasoning + Acting

Generates:

- Reasoning traces
 - Allows the model to track, update action plans, and even handle exceptions
- Task-specific actions
 - Allows to interface with and gather information from external sources such as knowledge bases or environments

Server
BE

LLM

ReAct: Reason + Act

COT



- Chain-of-thought (CoT) has shown that LLM has reasoning capabilities
- It still lacks access to the external world
- Unable to update its knowledge can lead to issues like fact hallucination and error propagation.

ReAct

- LLM generate verbal reasoning traces and actions for a task.
- Allows perform dynamic reasoning to create, maintain, and adjust plans for acting
- Enables interaction to external environments (e.g., Wikipedia) to incorporate

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Program??

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Program??

Apple Remote

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Apple Remote

Program??

Other
Devices??

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Apple Remote

Program??

Other
Devices??

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?



Apple Remote

Program??

Other
Devices??

ReAct: Reason + Act

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Apple Remote

Front Row

Search

Keyboard
Function Keys

ReAct: Reason + Act

(1d) ReAct (Reason + Act)

Aside from the Apple Remote, what other devices can control the program Apple Remote was originally designed to interact with?

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: Search[Apple Remote]

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: Search[Front Row]

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software) .

Act 3: Search[Front Row (software)]

Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: Finish[keyboard function keys]



Agents: Why do we need them?

For a question like

“What's the best time to visit the Grand Canyon?”

Although LLM has common knowledge. To answer this question it needs:

- Access to weather predictions and
- Tourism seasons
- Other factors that influence the visitor experience.
- ...

Agents: Why do we need them?

For a question like

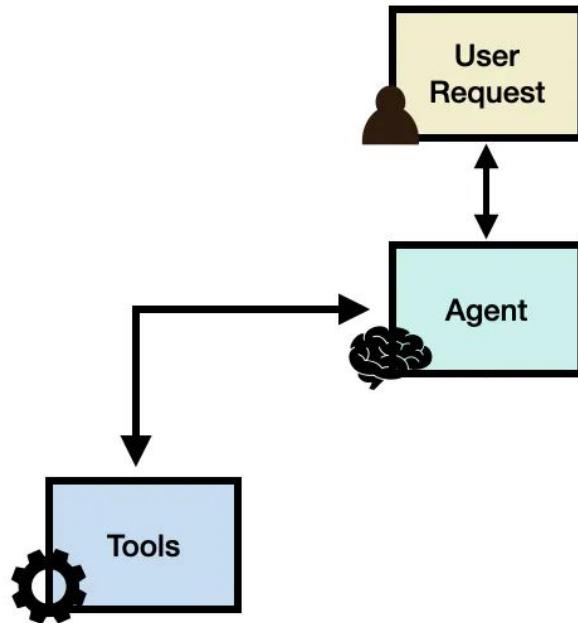
“What's the best time to visit the Grand Canyon?”

Although LLM has common knowledge. To answer this question it needs:

- Access to weather predictions and
- Tourism seasons
- Other factors that influence the visitor experience.
- ...

Tools

Agents with Tools



Agents with Tools

- Tools at the disposal of the LLM that can assist it in finishing a task
- How the LLM would choose these tools?
 - Which prompting technique help with that?
 - More on this later

Agents: Why do we need them?

For a question like

User: What are good places to visit in Arizona?

System: Grand Canyon is a very common attraction that ...

User: What's the best time to visit it?

System: The best time to visit Grand Canyon is

...

User: What is the average cost of doing that

...

What do we need to add?

Agents: Why do we need them?

For a question like

User: What are good places to visit in Arizona?

System: Grand Canyon is a very common attraction that ...

User: What's the best time to visit it?

System: The best time to visit Grand Canyon is

...

User: What is the average cost of doing that

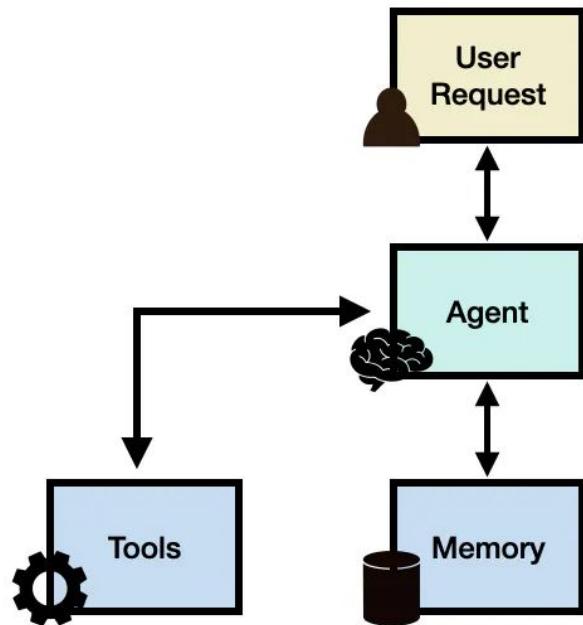
...

What do we need to add?

Agents: Memory

- Short-term memory
 - Context information about the agent's current situations
 - Realized by in-context learning which means it is short and finite due to context window constraints.
- Long-term memory
 - Agent's past behaviors and thoughts that need to be retained and recalled over an extended period of time
 - External vector store: scalable retrieval

Agents with Tools & Memory



Agents: Why do we need them?

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find `[Front Row]`. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search `Front Row (software)`.

Act 3: `Search[Front Row (software)]`

Obs 3: Front Row is a discontinued media center software ...

Thought 4: `Front Row (software)` is controlled by an `Apple Remote` or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



Agents: Why do we need them?

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find `[Front Row]`. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search `Front Row (software)`.

Act 3: `Search[Front Row (software)]`

Obs 3: Front Row is a discontinued media center software ...

Thought 4: `Front Row (software)` is controlled by an `Apple Remote` or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`

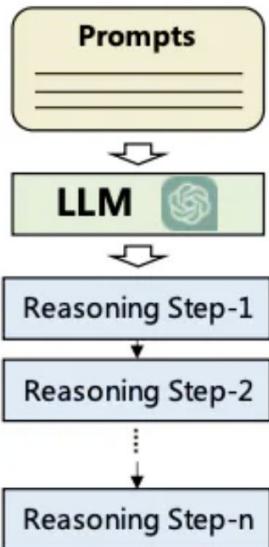


Planning

Agents: Planning

Planning without Feedback

CoT , Zero-shot CoT



Planning with Feedback

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find `[Front Row]`. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search `Front Row (software)` .

Act 3: `Search[Front Row (software)]`

Obs 3: Front Row is a discontinued media center software ...

Thought 4: `Front Row (software)` is controlled by an `Apple Remote` or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



Example Agent Frameworks

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: Search[Apple Remote]

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: Search[Front Row]

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

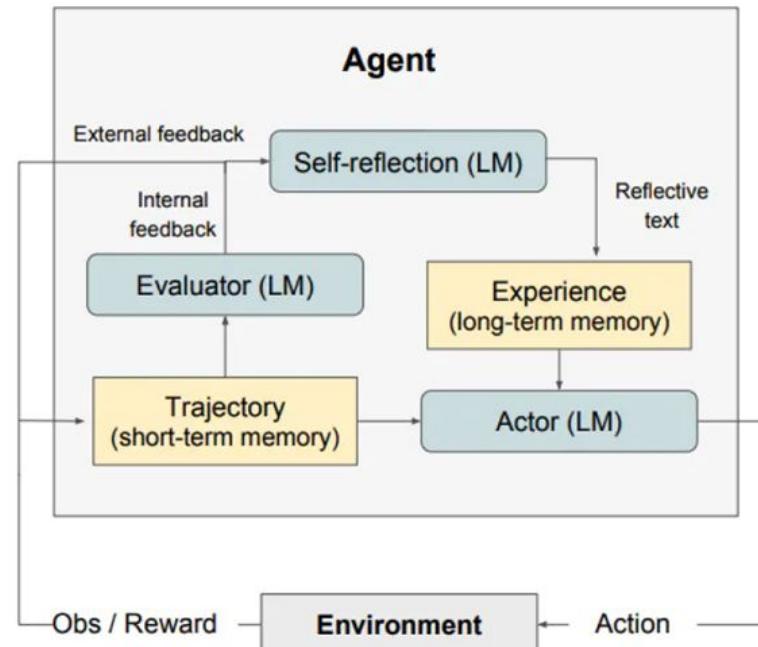
Thought 3: Front Row is not found. I need to search Front Row (software) .

Act 3: Search[Front Row (software)]

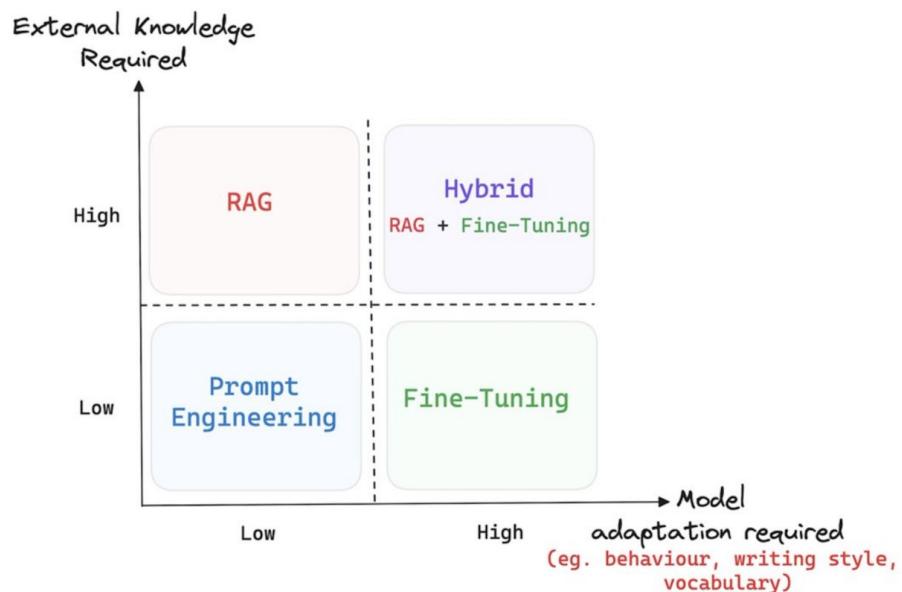
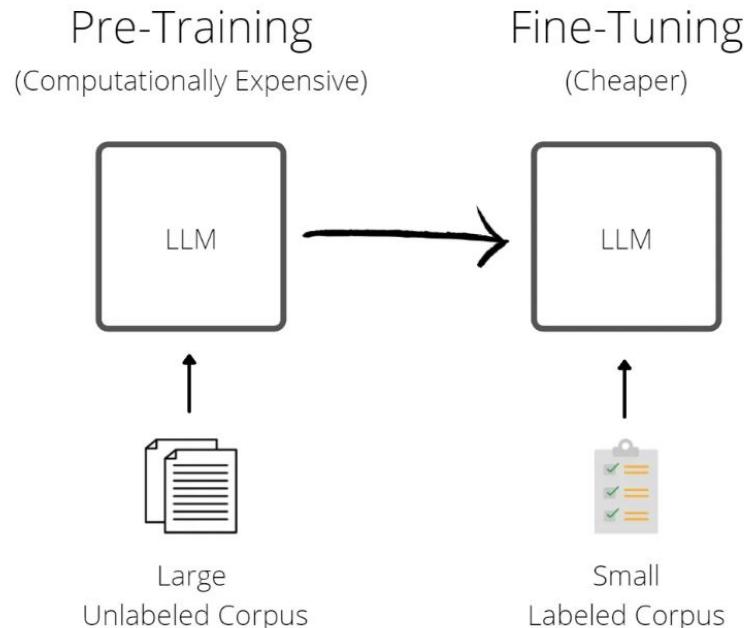
Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

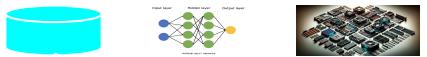
Act 4: Finish[keyboard function keys]



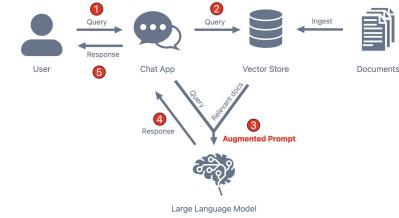
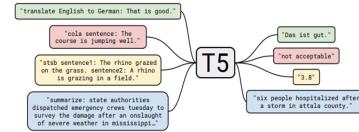
Why Not Finetuning ? Why is it hard



How Did We Get Here?!



Model



Requires Multitask
training

Requires Instruction
Following

Instruction Tuning

I am a
student

Soy un
estudiante

Translate the following from
English to Spanish:
“I am a student”

Soy un
estudiante

Instruction Tuning

- Now the model has been trained to follow instructions
- So, it generalizes to unseen instructions
- Sometimes people refer to this as **meta learning**
 - Or learning to learn
- Cool - we can do everything right once the model has been trained

<https://colab.research.google.com/drive/118F3pEZLVX-s0FiZJl7HfRz3tfPlfWiG>