Building applications on Top of LLMs

Codemash Precompiler session

BJ Allmon
Nilanjan Raychaudhuri

bjallmon@gmail.com
nilanjan@tublian.com

Logistics

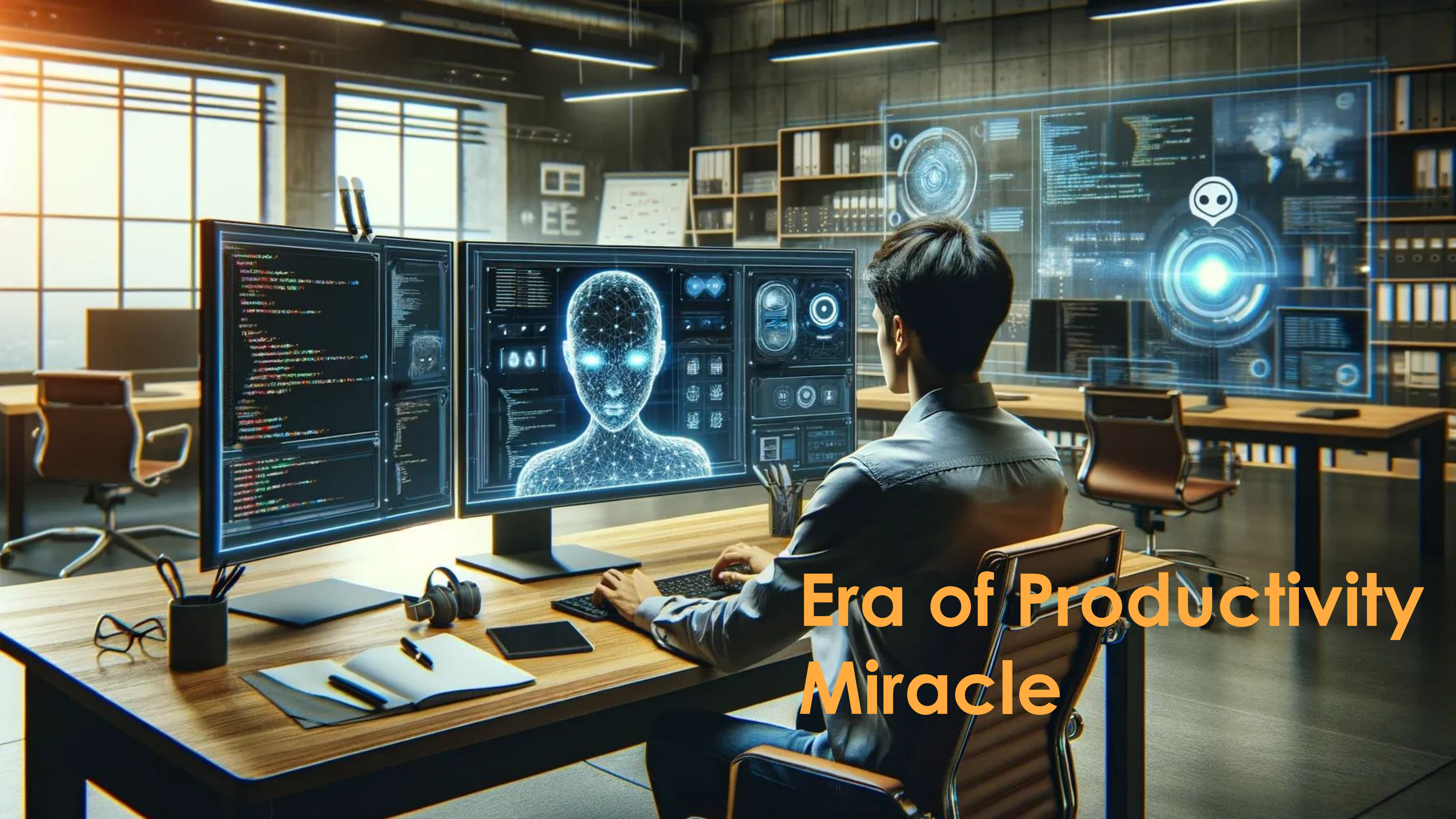
- 8am - noon, Part 1: Hello LLMs
- 12pm - 1pm, LUNCH BREAK
- 1pm - 5pm, Part 2: Build your own LLM app

Part 1

Hello LLMs

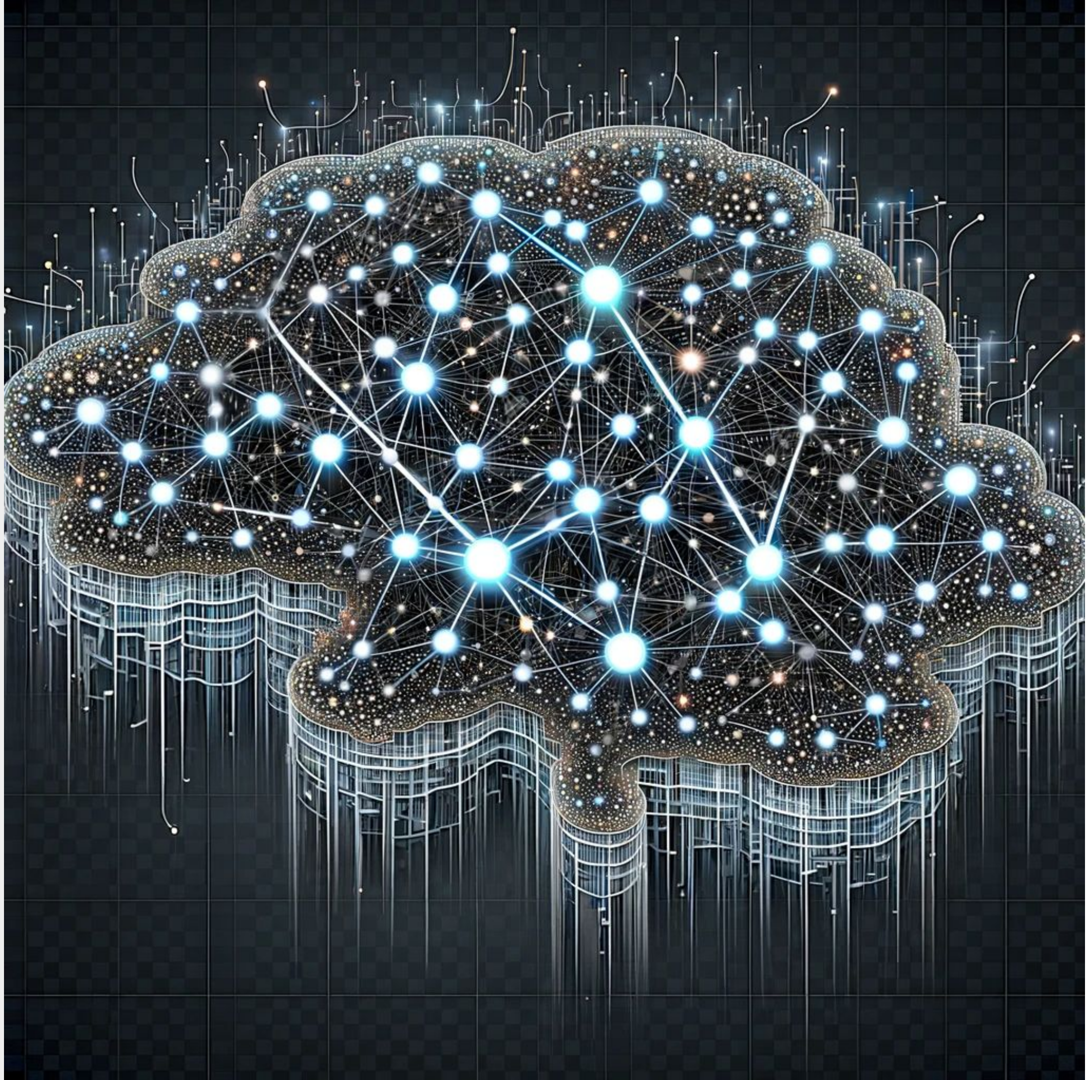
Part 1: Agenda

- Introduction to LLMs
- Look inside LLM
- What are LLM based applications?
- LLM TechStack
- Build our first application
- Introduction to RAG Architecture
- RAG in Action: Q&A Codemash Bot



Era of Productivity Miracle

Introduction to Large Language Models



Terms

Generative AI = Generate new content(text, audio, video, images etc) based on variety of input

LLM = Type of AI algorithm that uses deep learning and massive dataset to summarize, predict and generate new content

GPT = Generative pre-trained Transformer. A type of neural network that uses transformer architecture.

What is Language Model?

A language model is a machine learning model, that predicts the next word given a sequence of words.

Autocomplete is a language model, for example.

What is Large Language Model?

LLMs are intelligent pieces of code that can learn from a vast universe of data, make inferences using it, and use those inferences to answer a user’s questions or perform certain tasks.

History of Language Models

- Neural Networks and NLP Beginnings (1950s — 1990s)
- Statistical Language Models (2000s): n-grams and Hidden Markov Models (HMMs)
- Deep Learning Resurgence (2010s): Word embeddings (Word2Vec), RNN, LSTMs
- Introduction of Transformer Architecture (2017)
- BERT and Pre-training (2018)
- GPT(Generative Pre-Trained Transformer) Series (2018 — Present)

Key features of LLMs

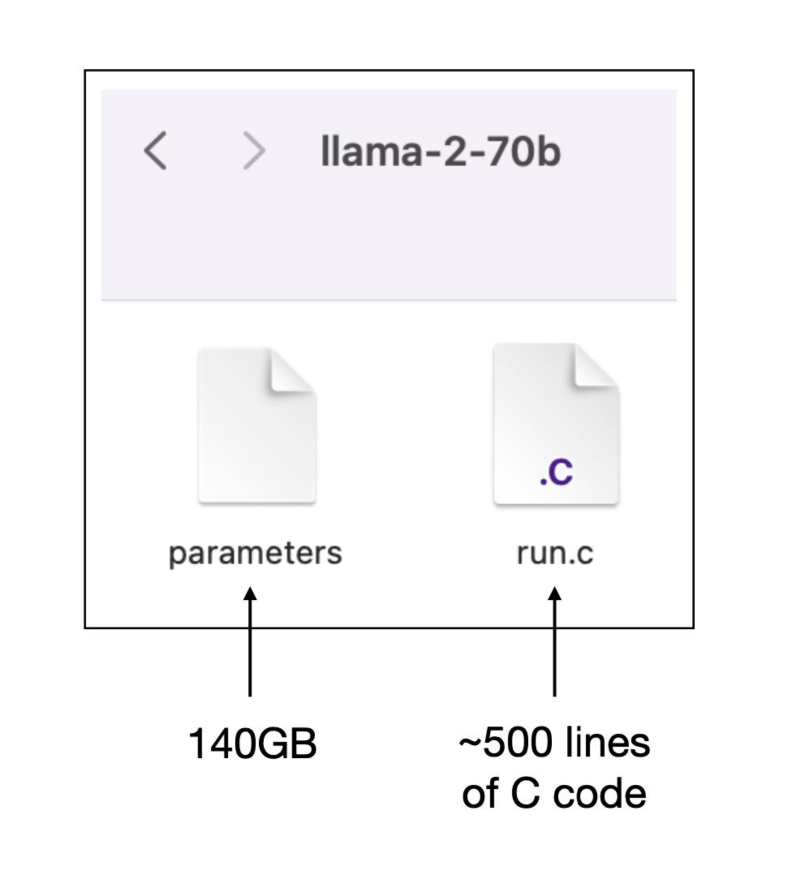
- **Neural Network Architecture - LLMs are based on neural network architectures, specifically a type known as Transformer models.**

- **Training Data & Process - LLMs are trained on extensive datasets comprising a wide range of text sources, such as books, websites, and articles.**

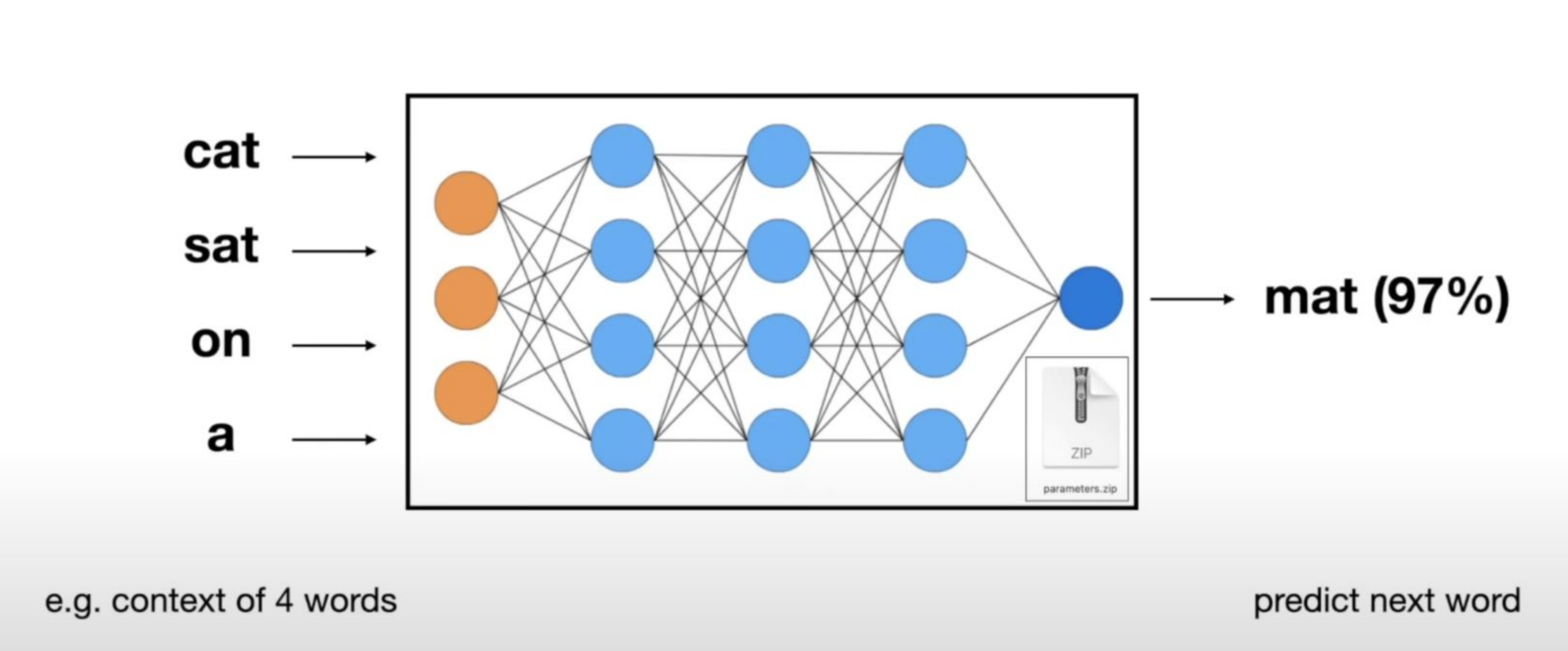
- **Capabilities - Once trained, these models can generate content(text, audio, video, images etc)**

Look Inside

Inside LLMs



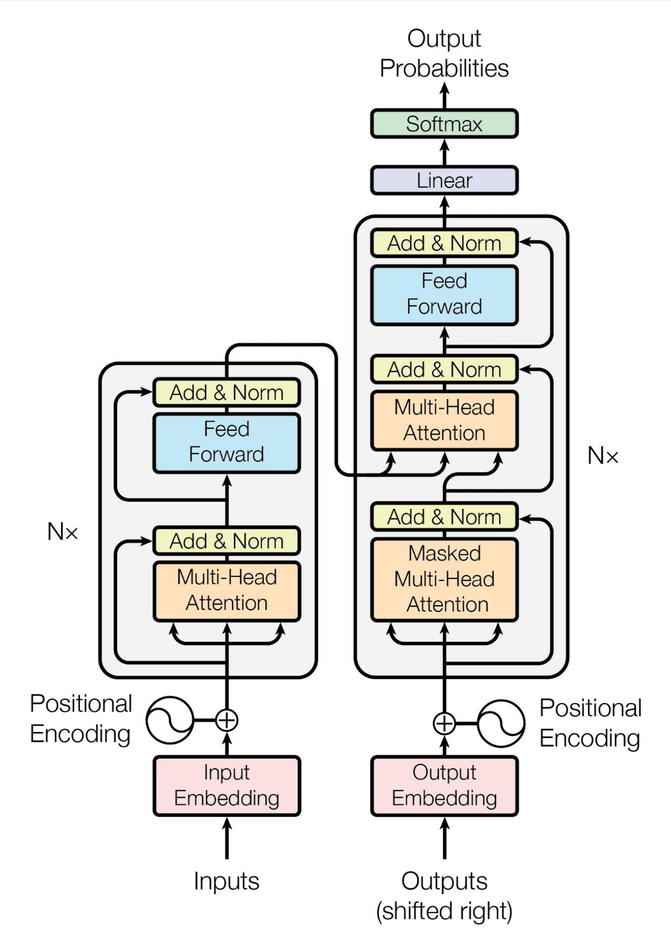
Neural Network



e.g. context of 4 words

predict next word

Transformer Architecture

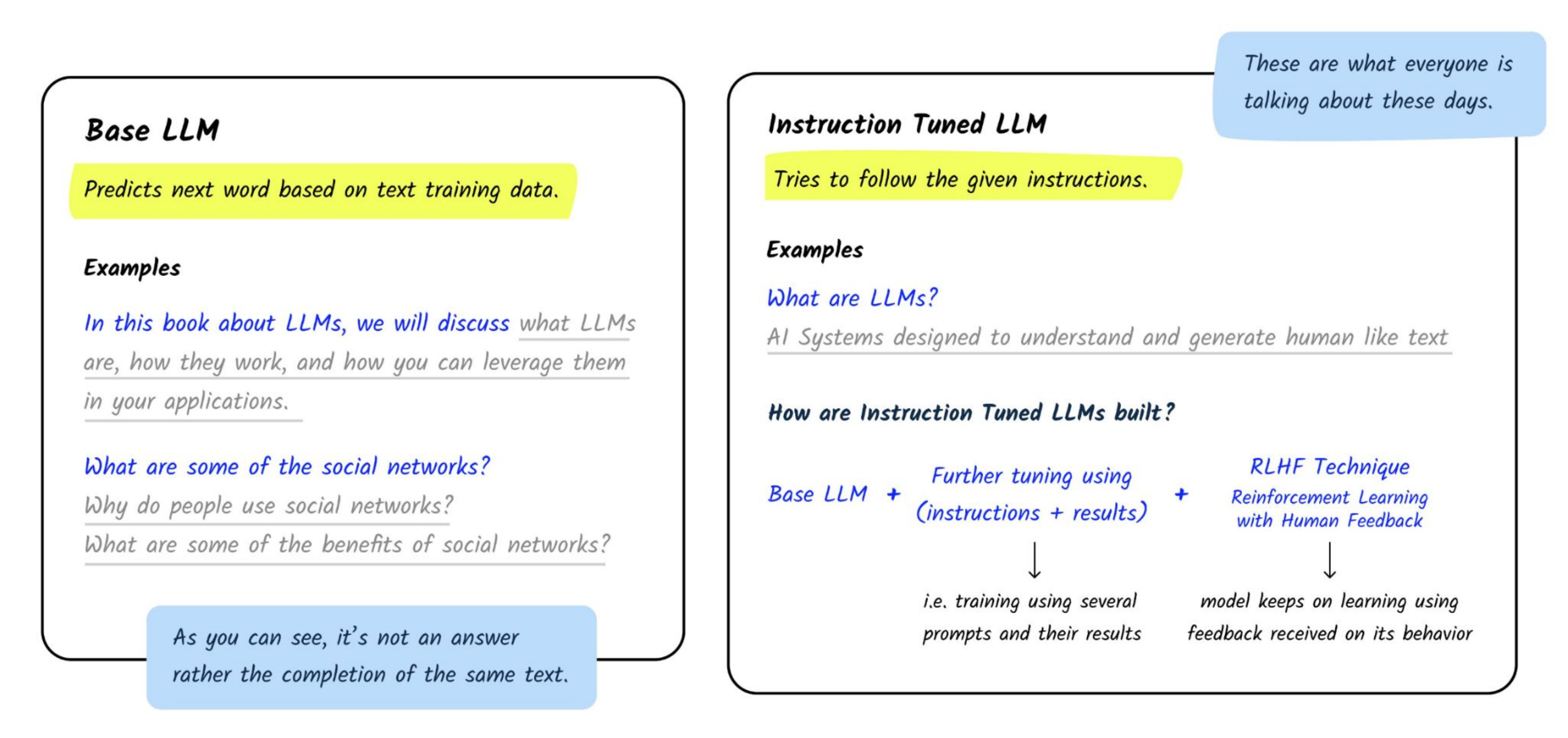


The transformer architecture is a specialized form of a neural network designed to handle sequential data like language more effectively.

Marketplace

https://machinelearningmastery.com/the-transformer-model/

Two types of LLMS



Base LLM

Predicts next word based on text training data.

Examples

In this book about LLMs, we will discuss what LLMs are, how they work, and how you can leverage them in your applications.

What are some of the social networks?
Why do people use social networks?
What are some of the benefits of social networks?

As you can see, it's not an answer rather the completion of the same text.

Instruction Tuned LLM

These are what everyone is talking about these days.

Tries to follow the given instructions.

Examples

What are LLMs?
AI Systems designed to understand and generate human like text

How are Instruction Tuned LLMs built?

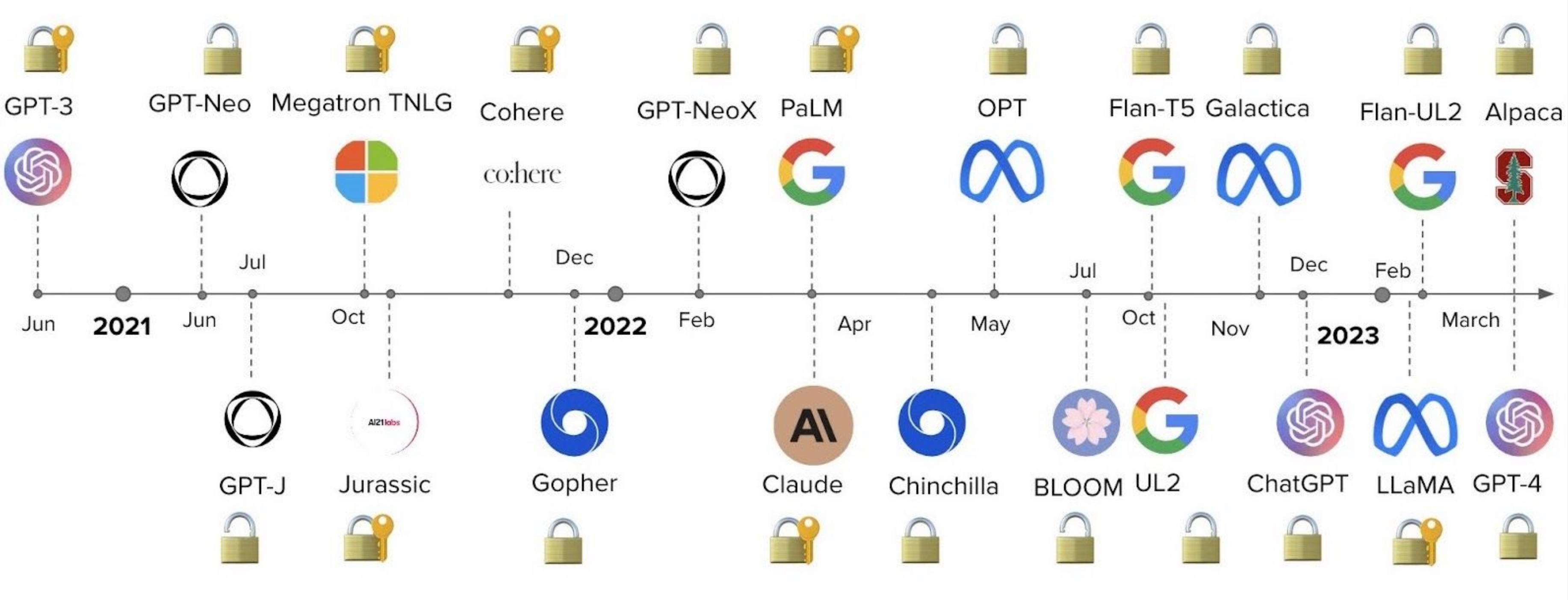
Base LLM + Further tuning using (instructions + results) + RLHF Technique Reinforcement Learning with Human Feedback

Further tuning using (instructions + results) ↓ i.e. training using several prompts and their results

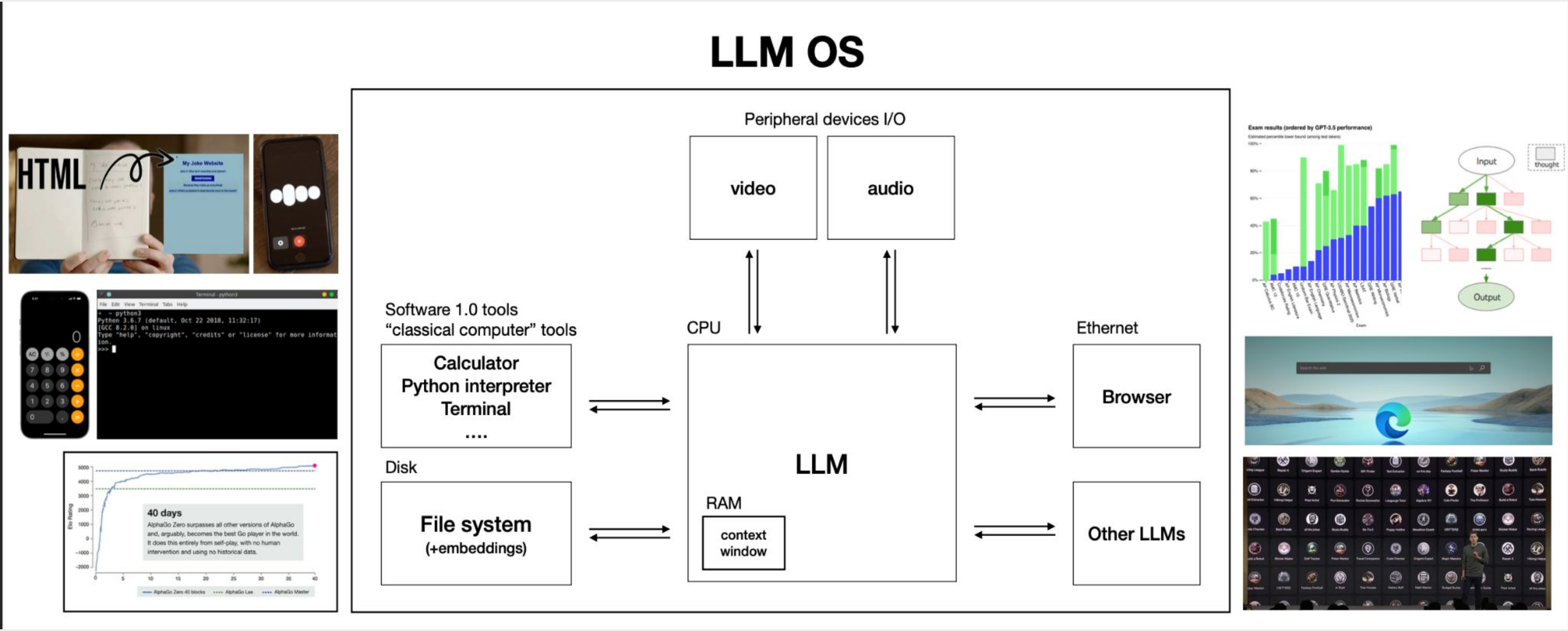
RLHF Technique ↓ model keeps on learning using feedback received on its behavior

Instruction Tuned LLMs = Base LLMs + Further Tuning + RLHF

Models


GPT-3
GPT-Neo
Megatron TNLG
Cohere
co:here
GPT-NeoX
PaLM
OPT
Flan-T5
Galactica
Flan-UL2
Alpaca
Jun
2021
Jun
Jul
Oct
Dec
2022
Feb
Apr
May
Jul
Oct
Nov
Dec
2023
Feb
March
AI21labs
GPT-J
Jurassic
Gopher
Claude
Chinchilla
BLOOM
UL2
ChatGPT
LLaMA
GPT-4

Think of LLM as a Operating System



LLM OS

Peripheral devices I/O

video

audio

CPU

LLM

RAM

context window

Software 1.0 tools
“classical computer” tools

Calculator
Python interpreter
Terminal
....

Disk

File system
(+embeddings)

Ethernet

Browser

Other LLMs

```
Terminal - python3
File  Edit  View  Terminal  Tabs  Help
→  ~ python3
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more informat
ion.
>>> 
```

40 days

AlphaGo Zero surpasses all other versions of AlphaGo and, arguably, becomes the best Go player in the world. It does this entirely from self-play, with no human intervention and using no historical data.

Prompt Engineering

Andrej Karpathy (from Intro to Large Language Models)

Limitations of LLMs

Applications on Top of LLMs

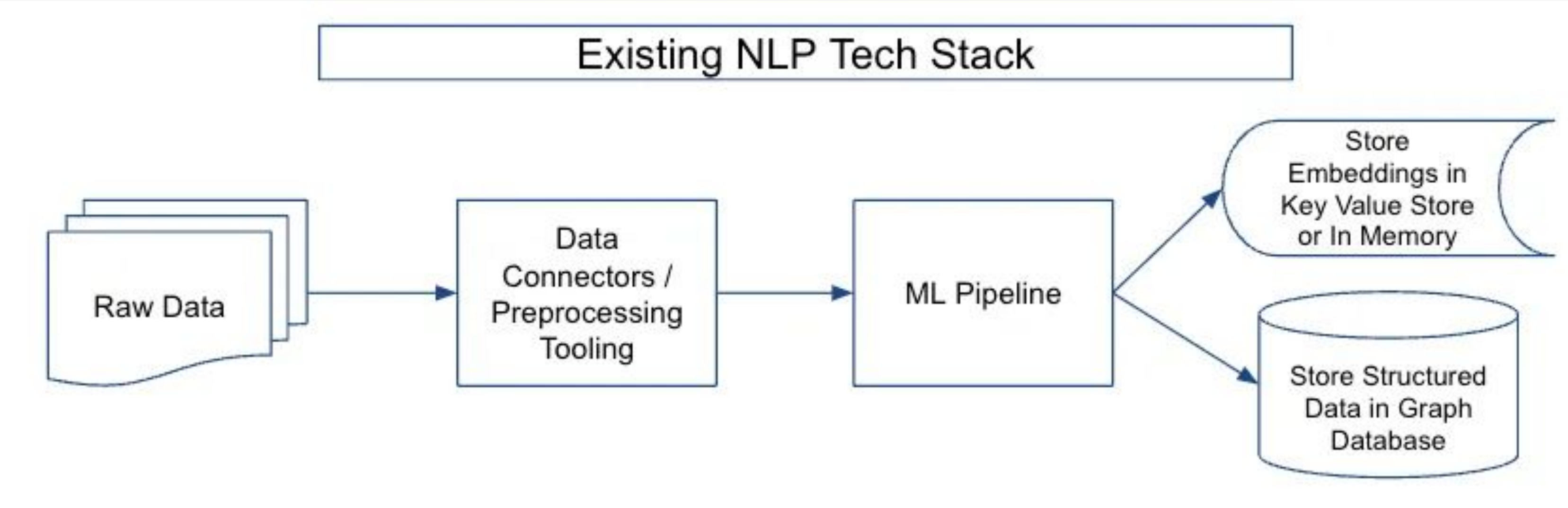
LLMs are specialized AI models used in various applications to understand, generate, and manipulate human-like text.

- ChatBots/Assistants
- Code Generation
- Sentiment Analysis
- Search Engine
- Legal/Healthcare/Education

Tech Stack

mission

Pre-LLM Architecture

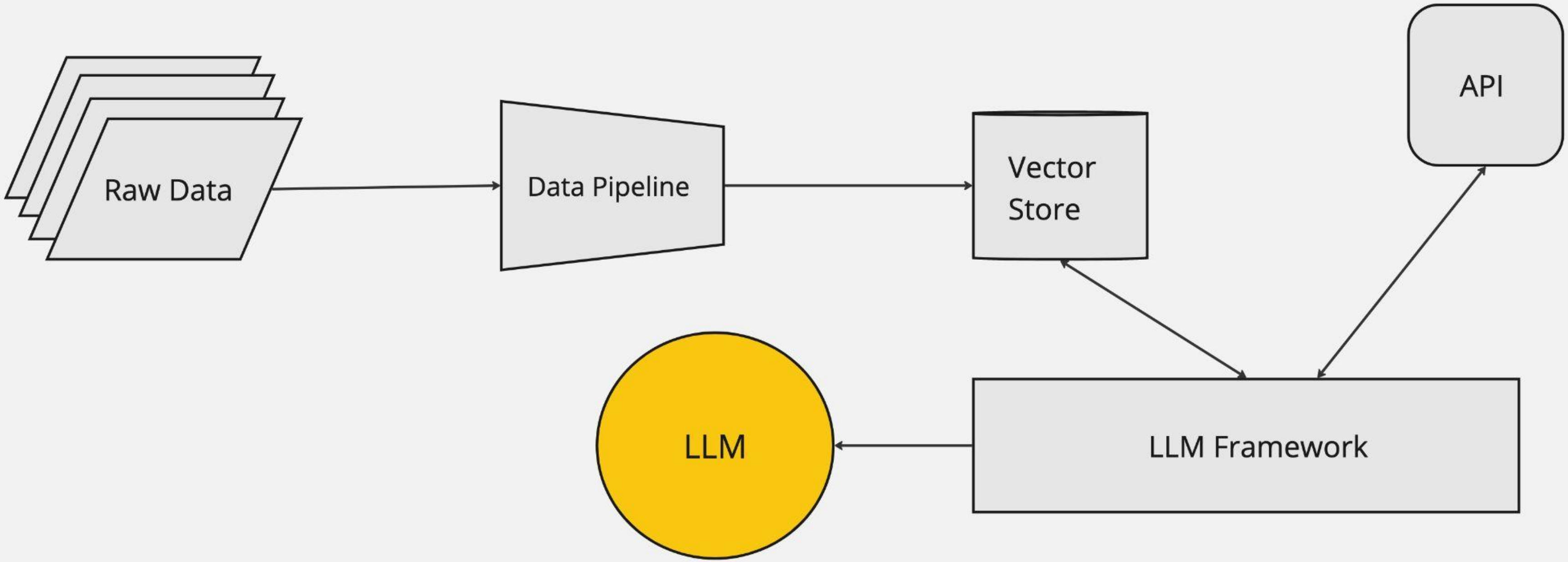

Existing NLP Tech Stack

Raw Data
Data Connectors / Preprocessing Tooling
ML Pipeline
Store Embeddings in Key Value Store or In Memory
Store Structured Data in Graph Database

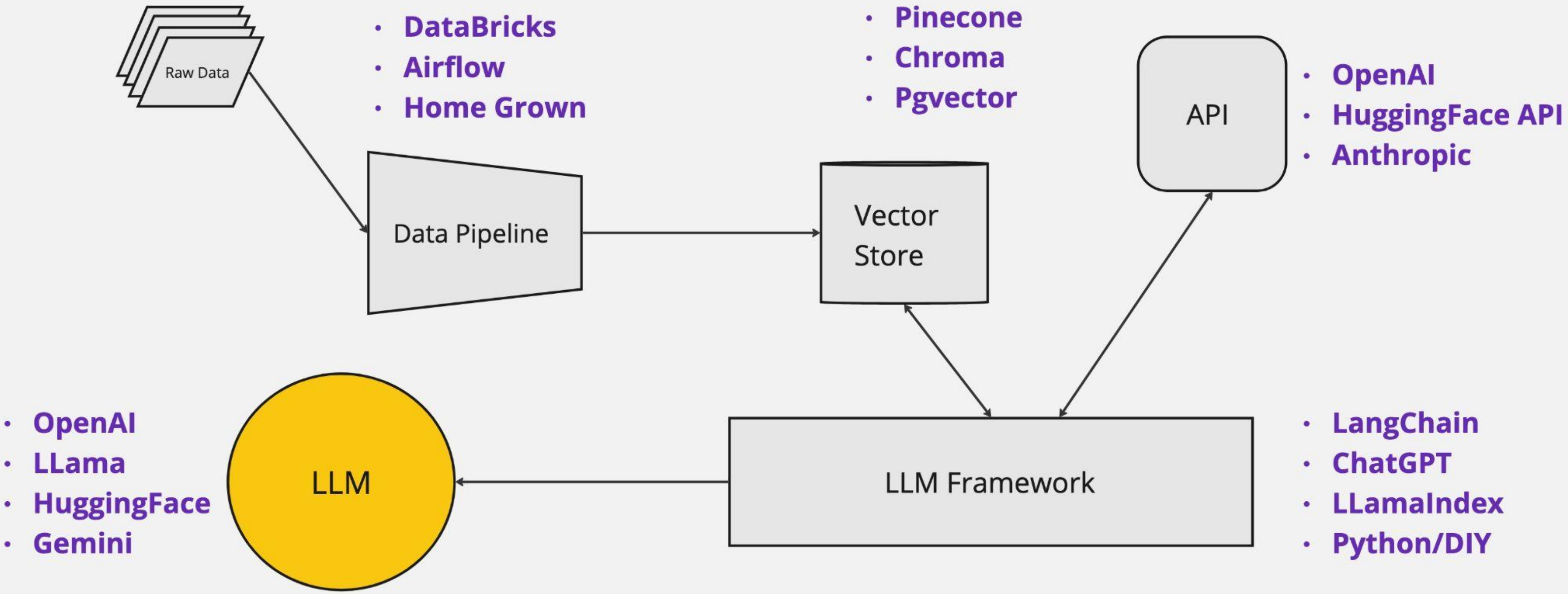
https://medium.com/@brian_90925/llms-and-the-emerging-ml-tech-stack-6fa66ee4561a

Hello World_

Emerging LLM architecture



LLM TechStack



- DataBricks
- Airflow
- Home Grown

- Pinecone
- Chroma
- Pgvector

- OpenAI
- HuggingFace API
- Anthropic

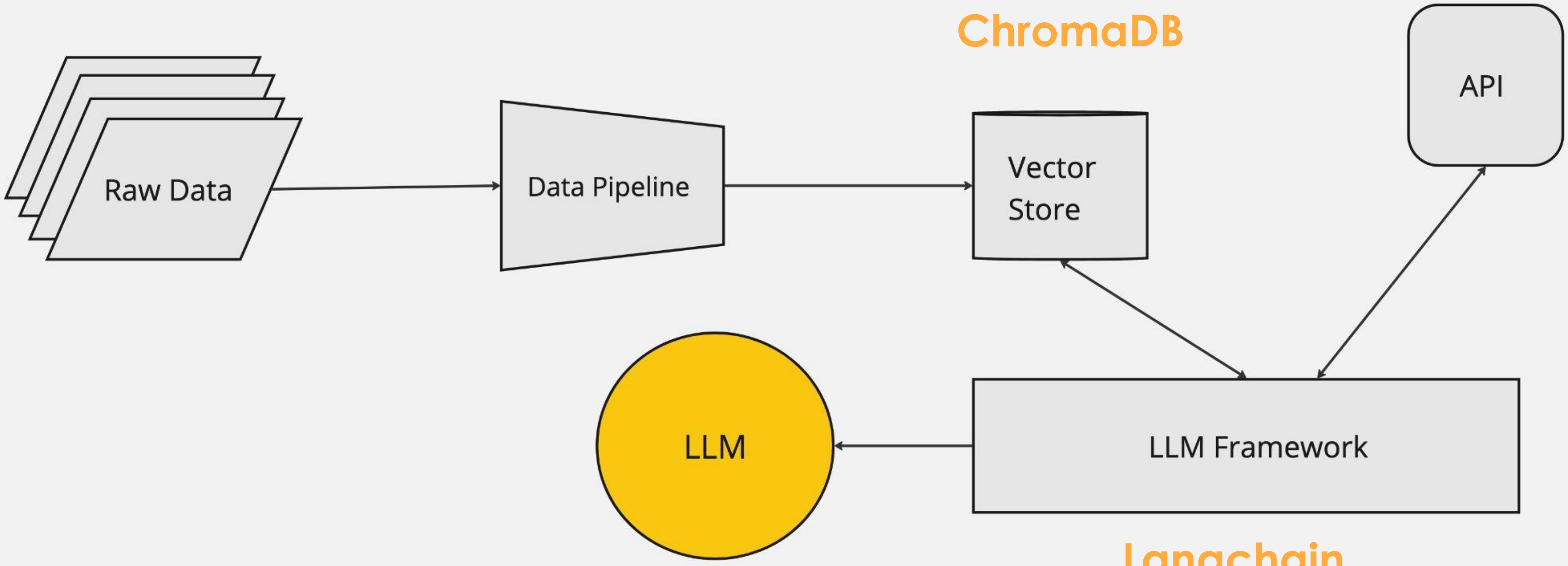
- LangChain
- ChatGPT
- LLamaIndex
- Python/DIY

- OpenAI
- LLama
- HuggingFace
- Gemini

Hello World_

Our Stack

Langchain



Langchain

GPT 3.5 Turbo

Hello LLM

Prompt Engineering

Sometimes prompt engineering feels like a new command line tool.

Sometimes it feels like asking an intern to do something and spending 10 minutes explaining step by step how to do it.

What is prompt engineering?

Prompt engineering is a new discipline for developing and optimizing prompts to efficiently use language models (LMs) for a wide variety of applications and research topics.

Prompt engineering skills help to better understand the capabilities and limitations of large language models (LLMs).

Elements of a Prompt

1. Instruction

   A specific task or instruction you want the model to perform

2. Context

   External information/background to steer the model for better performance

3. Input Data

   Input parameters/question

4. Output Indicator

   The type or format of the output

PromptTemplate

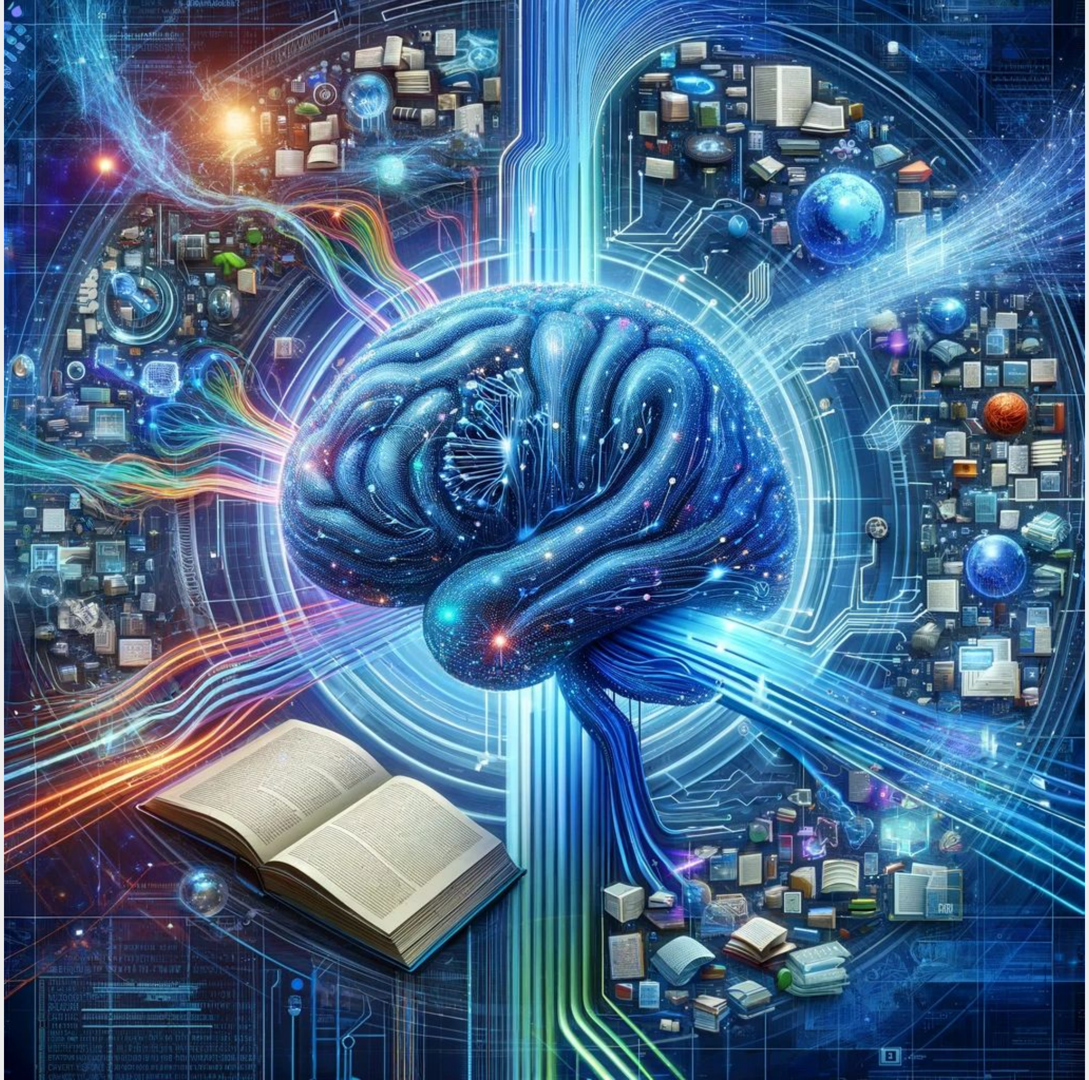
Prompt templates are predefined recipes for generating prompts for language models.

1. PromptTemplate

2. ChatPromptTemplate

   SystemMessage, HumanMessage, AIMessage

RAG



What is RAG?

Retrieval-Augmented Generation (RAG) is a technique for augmenting LLM knowledge with additional, often private or real-time, data.

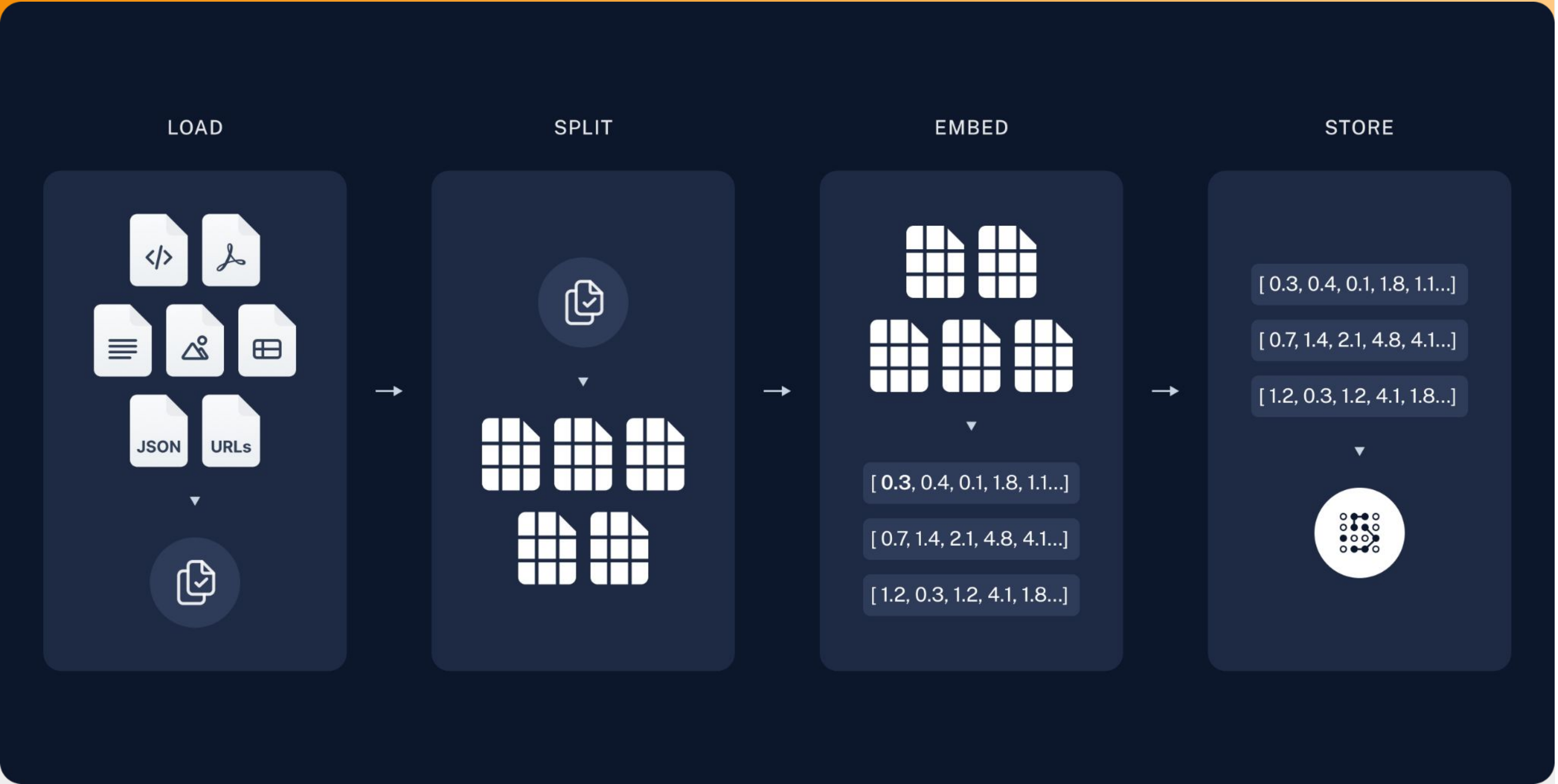
Why use RAG?

The key advantage of RAG is that it allows language models to provide more accurate, detailed, and current responses than they could using only the information they were trained on.

RAG Workflow

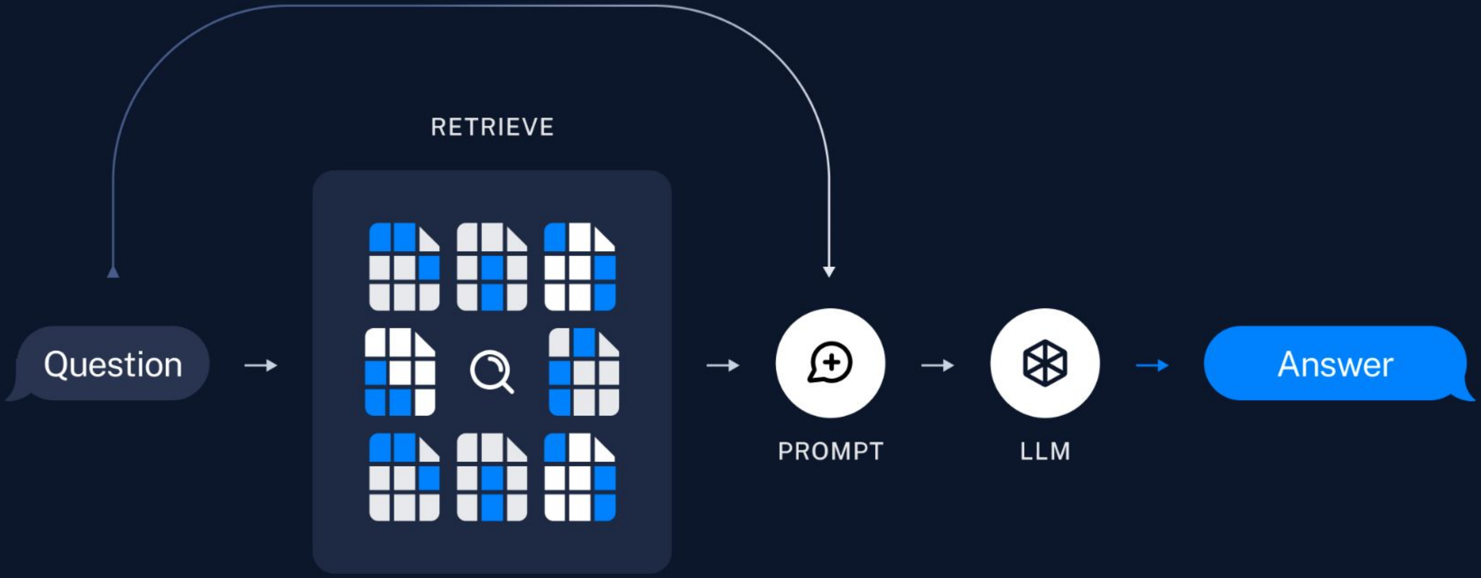
- Query Generation
- Data Retrieval
- Response Generation
- Refinement & Contextualization

RAG Architecture - Store



Marketplace

RAG Architecture - Query



Codemash Bot

mission

Part 1 Recap

- Getting started
- End-to-End with LLMs

Building Applications on Top of LLMs

LUNCH 12pm-1pm

Part 2

Build your own LLM app

Outcomes & Opportunities over Solutions

Avoid the Hammer of Thor for all solutions by starting with desired outcomes and opportunities. Consider all possible solutions.

Opportunity Map

EXAMPLE: Customer Technical Support

Desired Outcome
Improve tech support agent knowledge, retention, and problem-solving skills

Opportunity
Information Overload. Agents struggle to find relevant info quickly, leading to delays in issue resolution.

Solution
Develop a system that uses a RAG model to filter and present concise, contextual relevant information for specific technical problems.

Experiment
Build minimal agent RAG model and user test it with agents for speed, conciseness, and clarity of resolution.

Opportunity
Ineffective Learning Engagement. Traditional learning methods often fail to engage tech support agents effectively.

Solution
Interactive sessions where a RAG model actively engages agents in solving technical challenges, making learning more dynamic and engaging.

Experiment
Develop interactive learning modules with RAG assistance and evaluate engagement metrics to measure the effectiveness of the approach.

Opportunity
Lack of Personalization. One-size-fits-all training programs may not cater to individual learning preferences and needs.

Solution
Dynamic learning pathway system using a RAG model to provide personalized content recommendations based on individual agent performance and preferences.

Experiment
Integrate a personalized learning pathway feature into a LLM and assess its impact on agent satisfaction and learning outcomes.

Option 1:
Chat with a Book

https://www.tublian.com/challenge/303

Option 2:

Build your own LLM App

mission

Show & Tell

Part 2 Wrap up

- Productionizing LLM apps for the real world
- Next steps from here

Building Applications on Top of LLMs

Please leave us your valuable feedback on this session!

Thank You!

bjallmon@gmail.com
nilanjan@tublian.com