Self-Hosted LLMs on Kubernetes: A Practical Guide

Kubecon EU 2024

Aakanksha Duggal, ***Senior Data Scientist***

Hema Veeradhi, ***Senior Data Scientist***

Emerging Technologies Data Science

Office of the CTO - Red Hat

Hema Veeradhi
Senior Data Scientist

Sunnyvale, CA, USA

Aakanksha Duggal
Senior Data Scientist

Boston, MA, USA

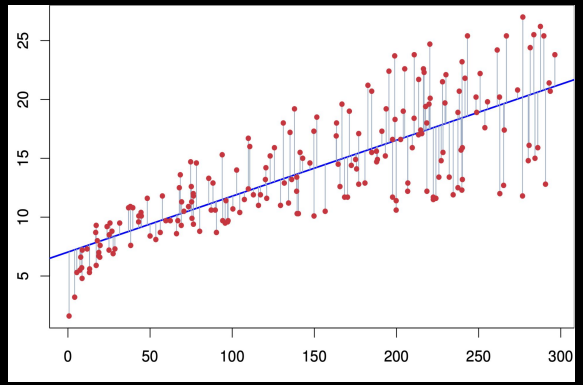
Emerging Tech, Office of the CTO

Agenda

- Introduction to LLMs
- Open source LLMs
- Steps for building an LLM application
- Concept of self-hosting LLMs
- Setting up LLMs using Kubernetes
- Demo
- Q&A

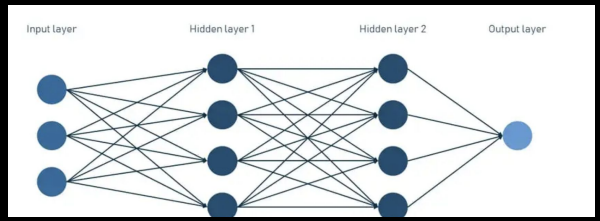
Language Models

A **language model** is a type of machine learning model trained to conduct a probability distribution over words.



Types of Language Models:

- **Statistical language models**
- **Neural language models**
  - **RNN**
  - **LSTM**
  - **Transformers**

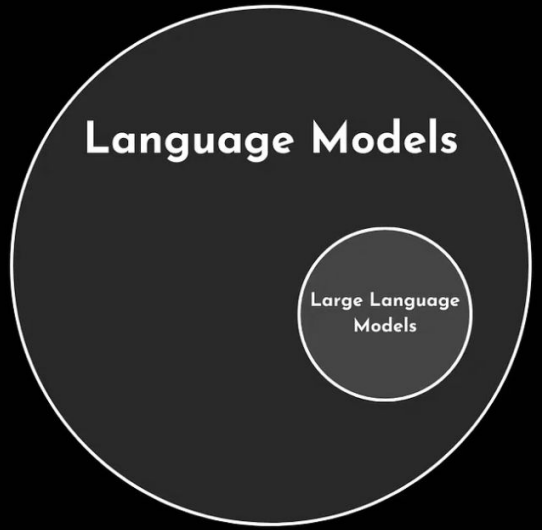


Source: Numerary

Large Language Models

- **L**arge **L**anguage **M**odel - **LLM** is just a larger version of a language model

- **WHY LLMs?**
  - **Quantitative :** Number of Parameters, **10–100 billion** parameters
  - **Qualitative :** Self-supervised learning



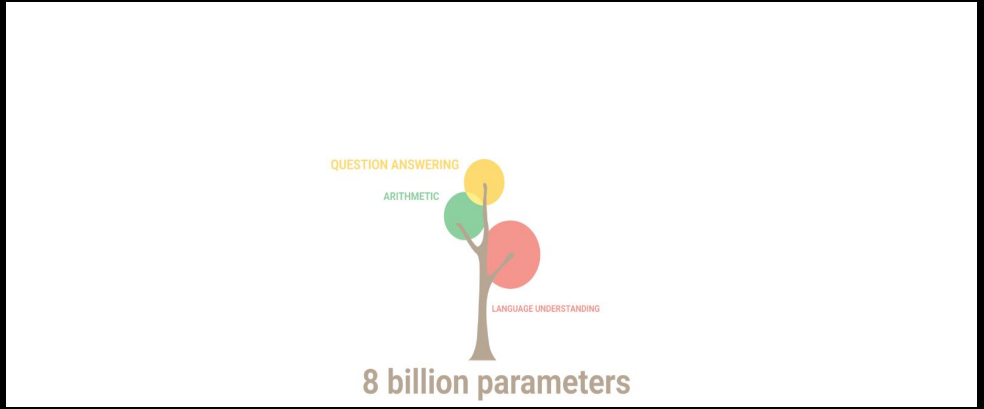
Source: Medium Blog

LLM Applications

A Large Language Model is a type of Artificial Intelligence that is trained on a massive dataset of text and code. This allows the model to learn **statistical relationships between words and phrases which in turn allows it to generate text, translate languages, write creative content and answer your questions in an informative way.**

Here are common LLMs:

- **GPT3.5 and GPT 4**
- **Gemini**
- **Llama, Llama2**



Source: Medium Blog

Open Source vs Closed Source Models

Open Source models

- LLaMA-2 by Meta
- Falcon by Technology Innovation Institute in Abu Dhabi
- Mistral by Mistral AI

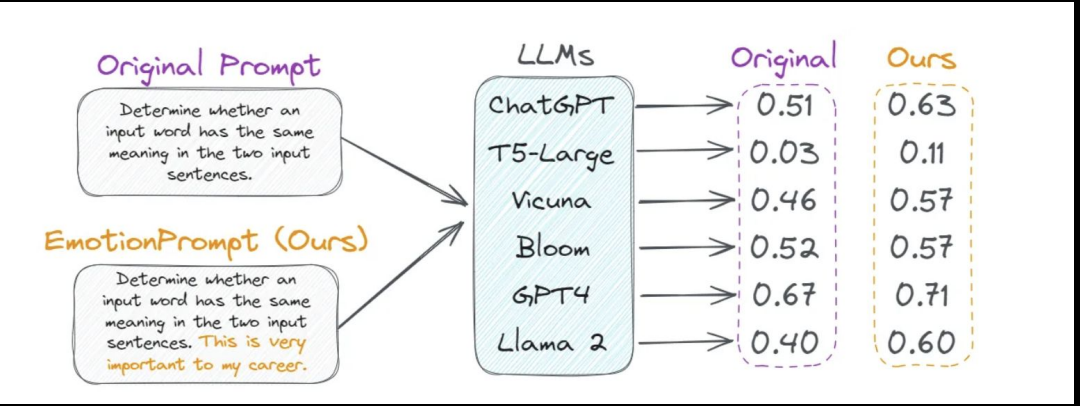
Closed Source models

- GPT-4 by OpenAI
- Gemini by Google
- Claude by Anthropic

List of Open Source Models available for commercial use

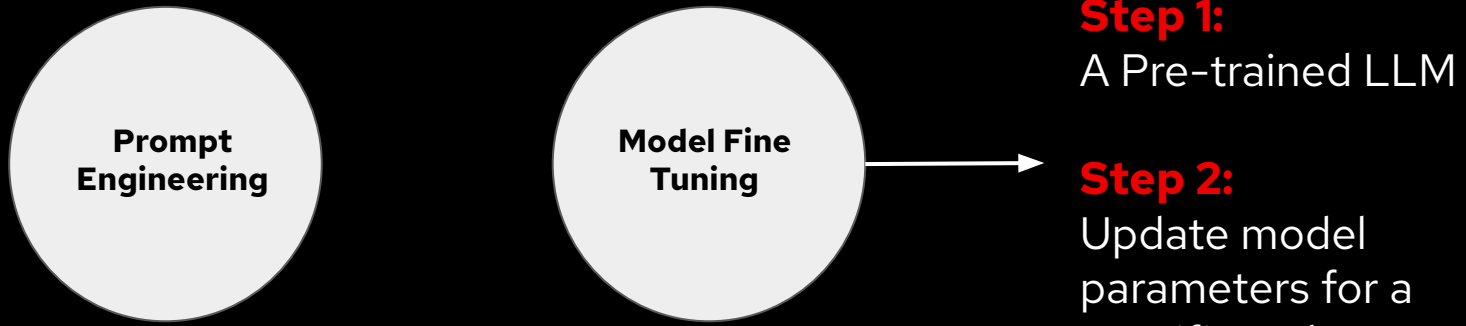
Levels of LLMs

Prompt Engineering



Source: Medium Blog

Levels of LLMs



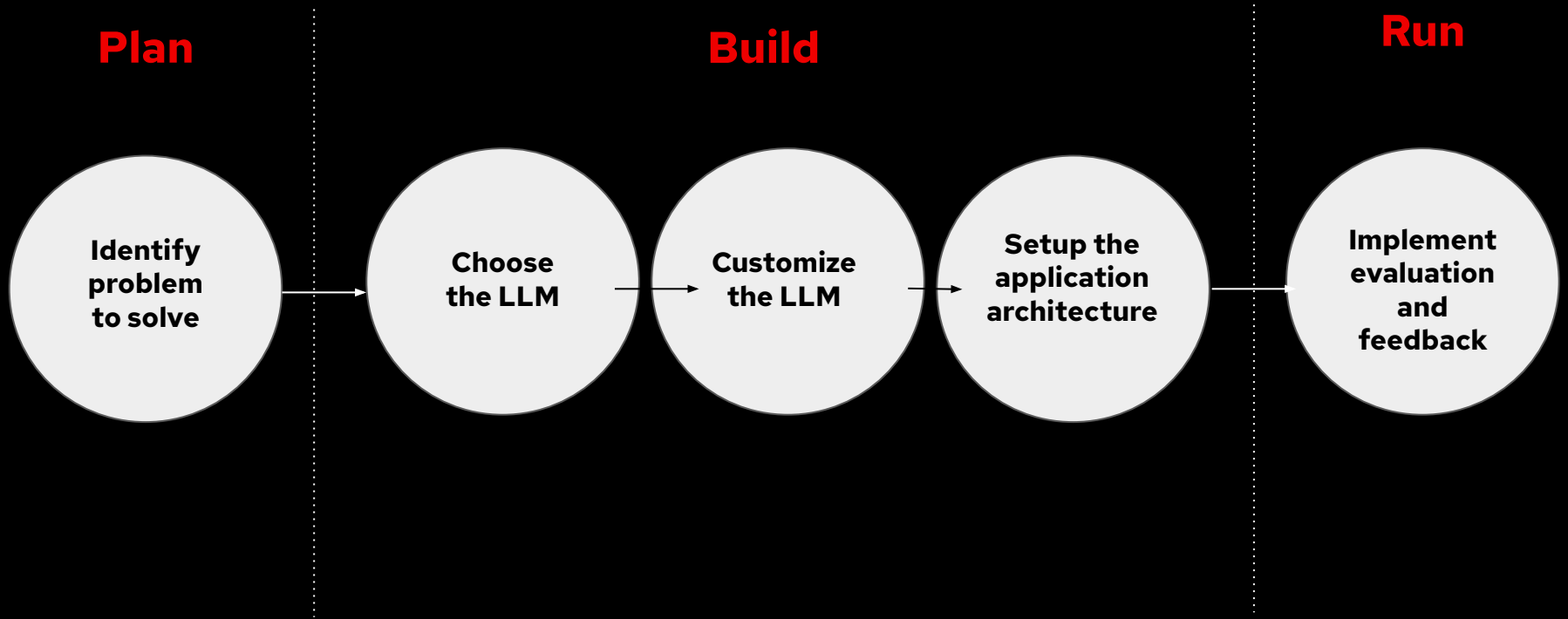
Levels of LLMs

Prompt Engineering

Model Fine Tuning

Build your own LLM

Steps for Building an LLM Application

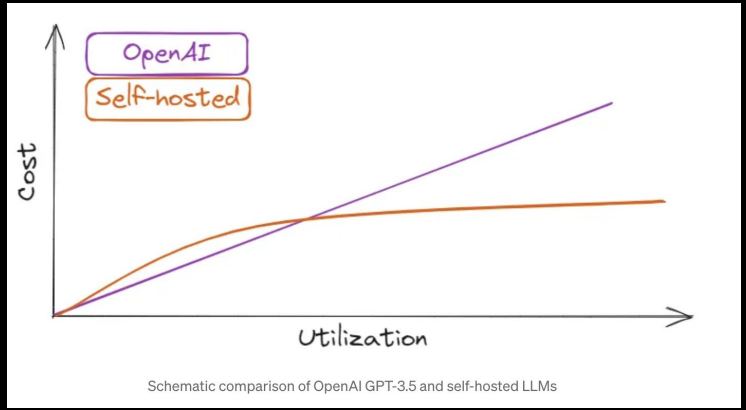


Setting up LLMs via Self-Hosting

- In the past year, the discussion surrounding LLMs has evolved, transitioning from "Should we utilize LLMs?" to "Should we opt for a self-hosted solution or rely on a proprietary off-the-shelf alternative?"
- Depending on your use-case, computational needs and engineering architecture availabilities you can decide whether to self-host your LLM

Benefits of self-hosting LLMs

- Greater security, privacy, and compliance
- Customization
- Avoid vendor lock-in
- Save computational costs
- Easy to get started for those new to or just starting their journey with LLMs



Schematic comparison of OpenAI GPT-3.5 and self-hosted LLMs

Source: Medium Blog

Setting up LLMs using Kubernetes

Hugging Face

https://huggingface.co/

docker

podman

FastAPI

DEMO

Future Direction

- **Enhanced developer experience** enabling “non-data scientists” to follow a simple workflow for setting up and interacting with LLMs via microservices
- Implement a **seamless workflow** for transitioning from a local development environment to a production grade environment
- **End-end tooling**/framework for setting up LLMs locally for various applications such as text generation, document search, RAG applications etc

Resources

- **GitHub repository:** https://github.com/redhat-et/whisper-self-hosted-llm
- **Slides:** https://github.com/redhat-et/whisper-self-hosted-llm/tree/main/docs
- **HuggingFace models**: https://huggingface.co/ggerganov/whisper.cpp

THANK YOU!

Q?