**06. Agents and Tools**

- Agents involve an LLM making decisions about which Actions to take, taking that Action, seeing an Observation, and repeating that until done.

- When used correctly agents can be extremely powerful. In order to load agents, you should understand the following concepts:

    - **Tool:** A function that performs a specific duty. This can be things like: Google Search, Database lookup, Python REPL, other chains.

    - **LLM:** The language model powering the agent.

- **Agent:** The agent to use.

**Is Rag and agent same?**

RAG going to be our custum data which is going to be static data or our private data which we are feeding during inference and do query on top of it, But agent will connects to real time API and retrieves value by connecting **SerpApi** , which is a real-time API to access Google search results.

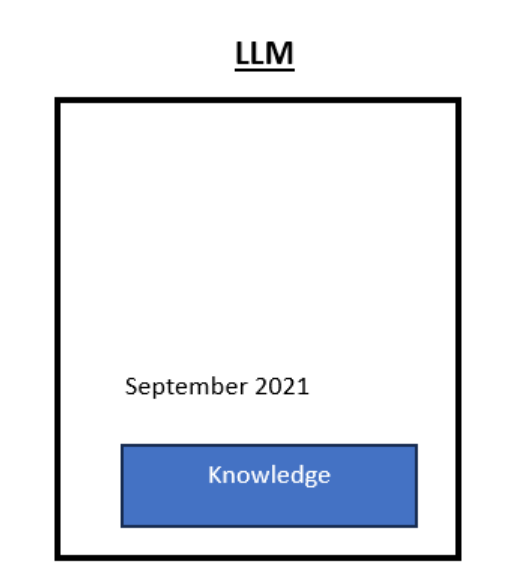
**Whenever we want real time data which is not there in LLM(Which trained using old data), then only we use Agents. If data is fixed then we use RAG**

**Agent is a very powerful concept in LangChain**

- For example I have to travel from Dubai to Canada, I type this in ChatGPT  
- Give me  two flight options from Dubai to Canada on September 1, 2023 | ChatGPT will not be able to answer because has knowledge till September 2021

- ChatGPT plus has Expedia Plugin, if we enable this plugin it will go to Expedia Plugin and will try to pull information about Flights & it will show the information

**What exactly happens when we try to enable this plugin**

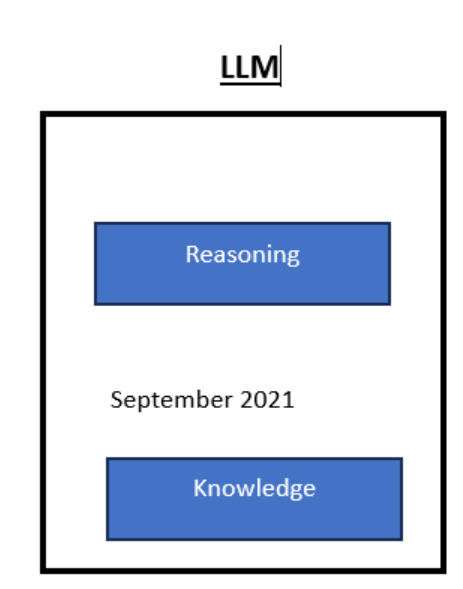


- When we think about LLM. Many people think that it is just a knowledge engine, it has knowledge and it will try to give answer based on that knowledge but the knowledge is only limited to September 2021. The think that most people missout is that Large Lanaguage Model has a reasoning engine, and using that reasoning engine it can figure out when someone types this type of Question

- Give me  two flight options from Dubai to Canada on September 1, 2023

- As a human we go to Expedia as we have a reasoning engine in our brain.

- LLM has a reasoning engine as well, so it will figure out the Source, Destination, Date and it will call Expedia Plugin and it will return response back.



**How much is US GDP in 2022? plus 5**

- As LLM has a reasoning engine to answer that question it will go to Google Search Tool, it will find that answer and then it will use Math Tool and do plus 5

**An agent has access to a suite of tools, and determines which ones to use depending on the user input.**

**#### Agent will conenct with external tools and it will use LLM reasoning capabilities**

- All the tools like **Google Search Tool and Math Tool are available as part of LangChain and you can configure  agent, so agent is nothing but using all these tools and LLM reasoning capabilities** to perform a given task

**To access Google Search Results in Real Time we use serpapi**

**serpapi and llm-math tool**

- **SerpApi** is a **real-time API** to access Google search results.

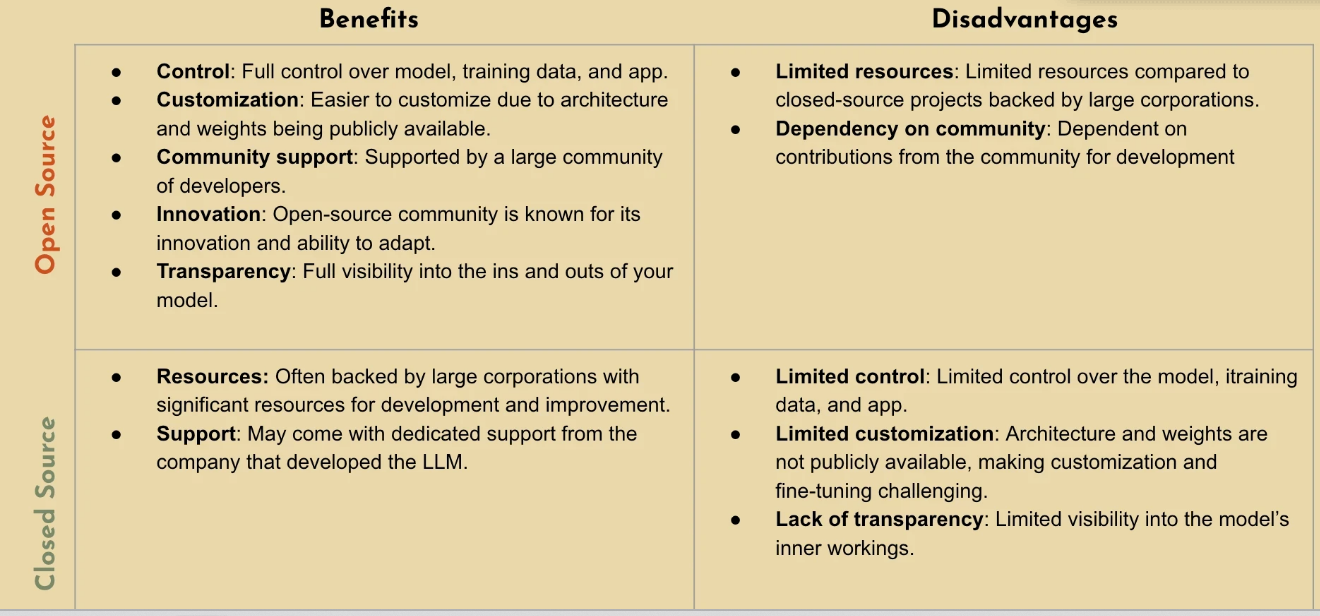
**https://serpapi.com/**

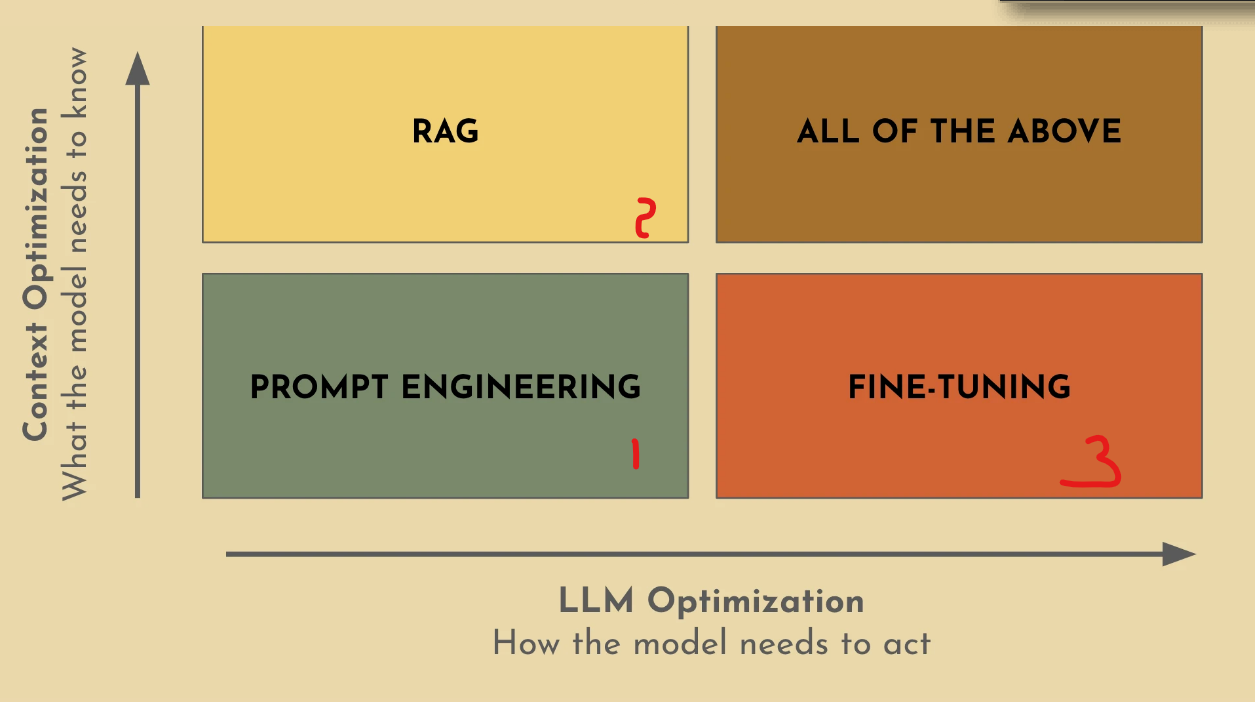
- If you're using a **text LLM**, first try **zero-shot-react-description**, aka. the **MRKL agent for LLMs**.

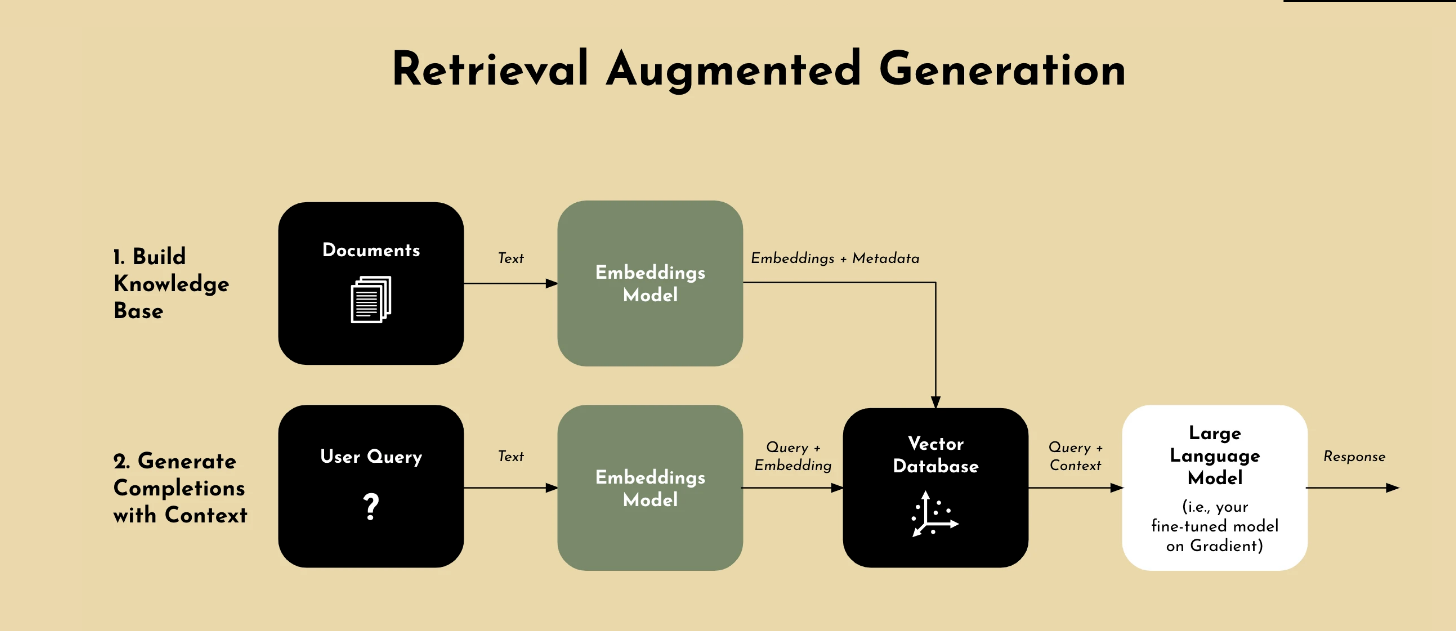
- If you're using a **Chat Model**, try **chat-zero-shot-react-descriptio**n, aka. the **MRKL agent for Chat** Models.

- If you're using a **Chat Model and want to use memory**, try **chat-conversational-react-description**, the **Conversational agent**.

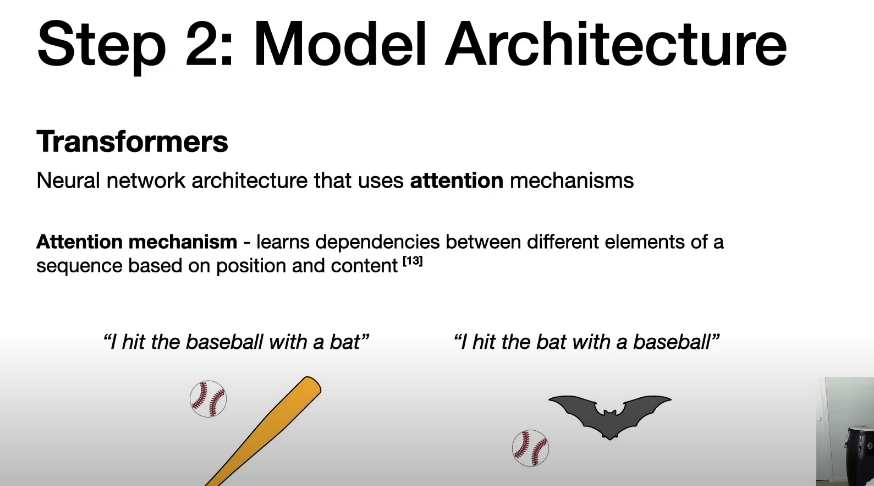
- If you have a **complex task** that requires many steps and you're interested in experimenting with a new type of agent, try the **Plan-and-Execute agent**.

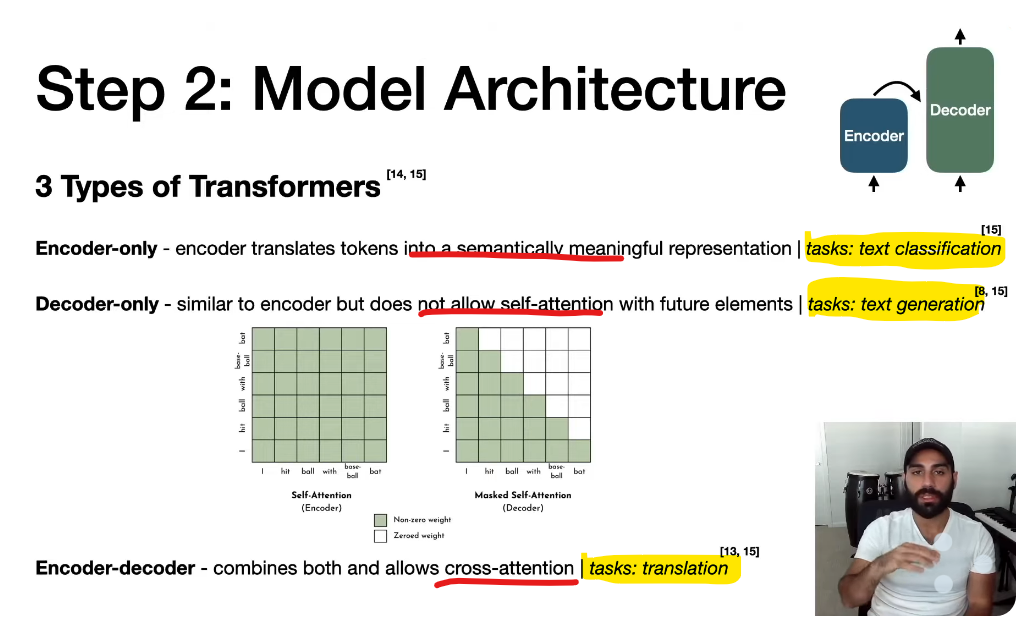


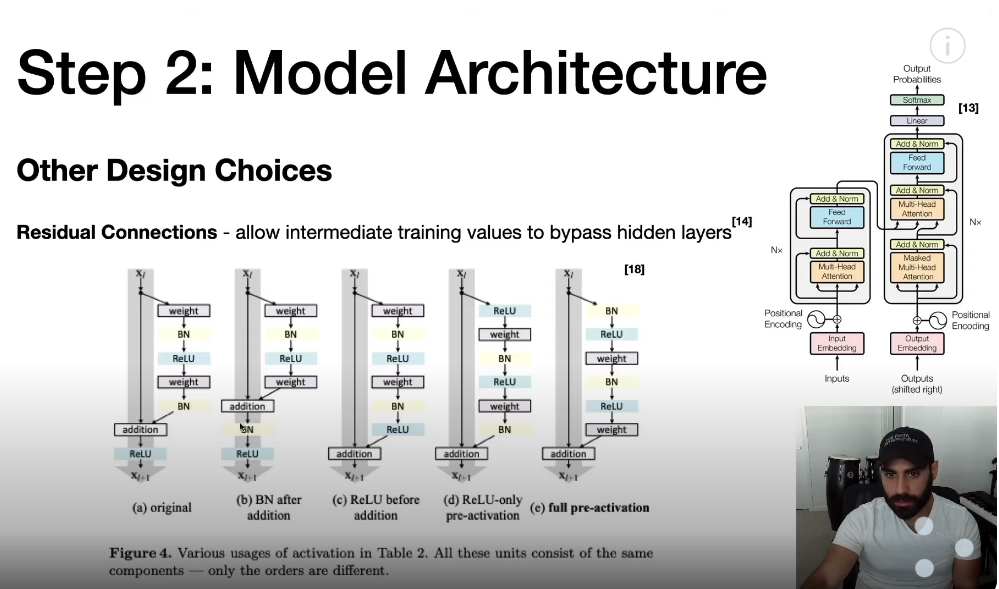


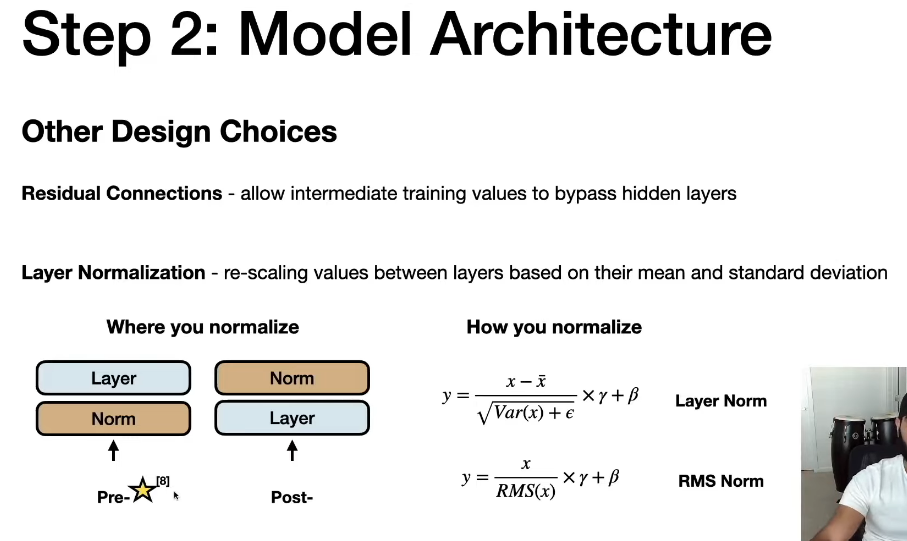


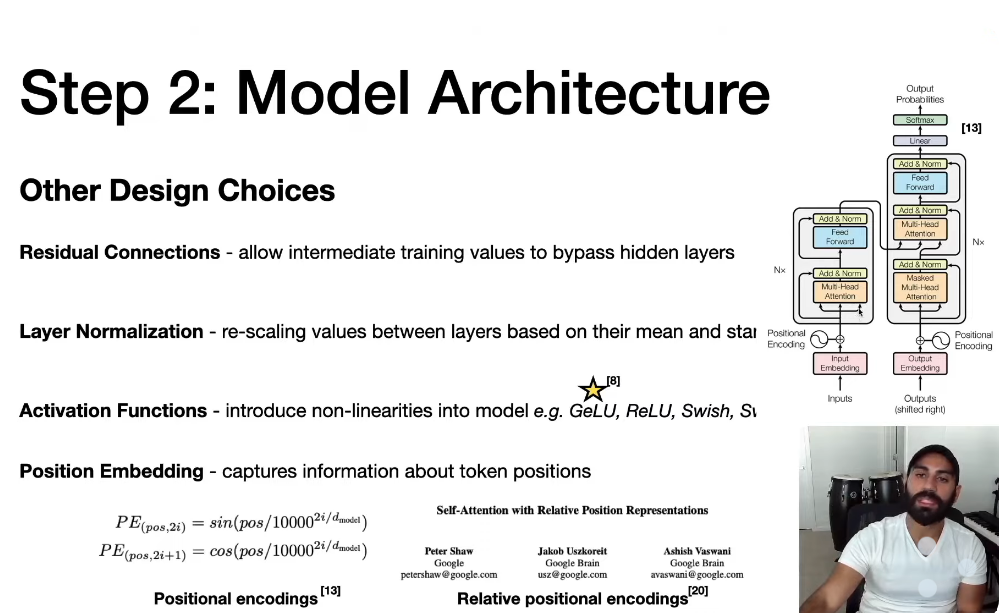
Transformer:

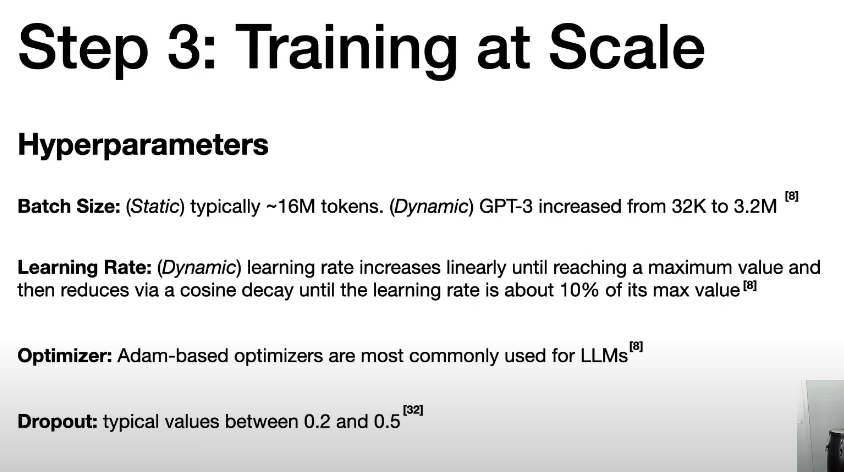












Fine Tuning:

3 way we can do.

* **Retrain all the parameter:** Update all the parameter of base model
* **Transfer Learning:** Freeze most of the parameter of base model and fine tune/update only last layer/few layer of base model’s parameter
* **Parameter Efficient Fine Tuning (PEFT):** Freeze **all** the parameter of base model and add **new trainable parameter** and fine tune that parameter . Ex: LoRA(Low Rank adaption)

