2.1 Transcript

Now let's talk about how we can program another LLM. LLMs mostly come with, APIs so that, you know, and third party libraries that you can then use to do whatever you want with it in your own programs.

So if you look at, the LLM that we have been using so far, this OpenAI's, OpenAI API that comes with an API of its own. Meta, that is Facebook's, LLaMa three is an API that's used to program meta and Google Gemini comes with the vertex API that you can use.

We're going to use, OpenAI's API because that's probably the most commonly used API right now.

And it actually turns out to be very easy and straightforward to use as well.

What does an API do? Like? Why do we need an API? Well, an API will help you, by, help will help you in writing complex threaded queries.

So typically when you're interacting with ChatGPT or with the chat version of, an LLM, what you can do, you know, you're going to give queries, get responses, get queries, get responses.

And you have to sort of think of how to focus your prompts in short, meaningful, statements, so to speak.

But if you have a very complex query with multiple questions and you want to sort of keep, a thread of the questions as you ask them, and as the responses come in, it becomes a little bit more complicated when you're dealing with a chat bot.

Instead, you can use an API to interact with the with the chat bot essentially, and or rather with the knowledge base, the, the large language model, you can interact directly with it through code.

And that code sort of keeps track of what you're doing and also is reproducible, you know, which is kind of nice, right?

It can also help because what you can do then as we've done that is you can extract relevant responses.

So when you get a response back from a chat bot and you want to save it, you're gonna have to cut and paste it or print the whole document PDF it or something like that, but you can't use it directly, right?

So if you want to use the response somewhere else, then you can actually extract that in your code and maybe attach it to another piece of code and do something with it.

You know, that's the idea here. So it becomes some more usable, you can use it to build code, you know.

So you one of the great things that LLM do is they, they are a great help in writing code.

They can actually write pretty good code, like open OpenAI's, GPT 4.0 for example.

It's hard to, for even medium complexity, code questions.

It comes out with correct code. You have to always check it. But, you know, it does a pretty good job.

So it's is very useful for building code. And when we we'll see some of this as we work with the OpenAI platform, API platform. Then this is the really the big thing.

You can tailor an LLM to your specific domain.

So for example, if you're our company that's selling a product and or an airline or, you know, Macy's or whatever, and you want to build a customer service agent using the LLM.

You want that customer service agent to respond to customers within the specific domain. In the in the case of an airline, it should respond with maybe alternate flights, or it'll give a coupon if the other one has planes that got canceled or something, you know, give a coupon or do something that is very airline specific.

In the case of Macy's, it wants to be able to talk about its products and say, hey, you know, you bought a TV and you want to return it, then this is how you return it and that kind of stuff.

And give the very specific to Macy's instructions on how to return your TV.

So typically when you use an LLM in a domain, you want your, the LLM to respond with everything that has to do with that specific domain and only respond in general terms when it doesn't have an answer from that domain.

So tailoring an LLM to a specific domain becomes a very important way in which you can use LLMs.

We saw a little bit of this when we were looking at, our interactions with ChatGPT, and we saw the custom, GPT is that we saw over there for specific things like logo generation, marketing plans and those kinds of things.

So these are all tailored LLMs. Somebody has done the tailoring for you, but you can do the tailoring for your own domain.

And it turns out to be not as hard as we think, as long as you can use the API.

And finally, you can use the API to embed them in your own application.

So you've built a customer service agent, your Macy's. You build a customer service agent.

You want it to be part of your web application.

The web front that a customer sees, right. So it needs to be embedded inside your application.

And you want to be able to do that. To do that, you need to use an API.

We're going to look at OpenAI's, API, which allows you to programmatically access various GPT models.

So the, you know, ChatGPT is built with many different models underlying it, there's a free GPT 3.5 and now, for all model that's a free to use with ChatGPT.

And then there are various 4 models, and then there probably be a five and a six and a seven as, ChatGPT gets better and better as the GPT model get better and better as they learn more.

So it, allows you to programmatically, access all of these models.

And what can you do with it?

You can do essentially what we said you had the API, you can interact with the models, and here you can ask questions, you can chat with the model.

You can analyze data, input data into the LLM and get responses, and you can get programming suggestions.

That is, to fix code or to write code or do any coding activity help you with any coding activity that you want.

You can build a specialized version of the LLM for your own application organization with an API, and you can embed a GPT based chat bot in your own web or mobile application.

And this would be a seamless integration of the LLM into your broader application, so that the user doesn't necessarily, is not sort of going saying, hey, I'm using an LLM you know, they just know that they have a chat bot and the chat bot is doing hopefully another know for sure, human like interaction, meaningful human like like interaction.

Because often human interactions are not meaningful as well, but a meaningful human like interaction with you, on the, on the web application.

And these are the kind of things that we want to see as we progress with this class.

We're going to see how we can interact, build and embed LLM based things inside our, inside our universe, so to speak. Before we actually move on to using open AI

We need to do make sure we have all the tools that we need. You will need to use an open, to create an OpenAI account.

So to create an open account, you go to this website over here and follow the instructions.

Then you create an account. Once you create an account, open AI requires you to create a project.

So the idea there is that whenever you use the API, you assign a particular API key to a project.

And you might actually be doing multiple things with the API.

So you might have a custom, chat bot that you built.

You may have a data analyst, analytics team that is also working on it, you know, and you have a single account, and that single account, allows you to do multiple things.

And for each thing, you're going to have your own secret, you know, its own each thing that you do with it, it's going to have its own secret API key.

So you create a project, and the project is going to contain the key that you can then use in your whatever you're doing, data analysis or building a chatbot or, you know, just playing around with that then you want to make sure you store the key in a safe place, right?

So the the idea here is that you don't want to expose your key in your code because your code is generally going to be shareable.

You show it to other people, and you don't want to share the key with other people because it's it's going to cost you, right.

Every time you use the OpenAI, API, you're going to pay a small fee to OpenAI, and you don't want that somebody else to take the key and then use it.

And, you run them a big bill and you're stuck having to pay it.

You don't want that to happen. So the, for this class, what do you want to do?

The safest, the easiest thing to do, actually, and if you're using Colab notebooks, is to go to your Google Drive and look for a folder called Colab notebooks, because that's where all your notebooks are going to be.

Right? And in that Colab notebook, just create a file, long ordinary text file, and pop the key inside that and then save it so your key is saved.

And in your code you're going to read that file and use the key without exposing it inside the code.

Okay. That's the idea here.

If you're using it locally, if you're not using Google Colab and you're using a local Jupyter notebook or some other methodology, then put it, you know where it makes sense to you. Easiest would be in the same directory or folder where your code lives, right?

That's if it's right there, then you don't have to, you know, deal with parts and all that kind of stuff.

You can, easily load it into your code.

So now let's take a look at how we can use the key to actually build stuff, you know, and then we'll work on a notebook that's going to show us how we can use the OpenAI API.