Python and AI Power-Up Program Offline Class-20250904_113631-Meeting Recording

September 4, 2025, 6:06AM

1h 49m 55s

Ajay Patel started transcription

Tarun Jain 0:11

Uh, share my screen.

So what we can do is we can create a new collab notebook. So here we can do 2 process. Either we can directly use VS code. So if you're using VS code, what you can do is you can create 2 files.

The first file is ingest.py. In ingest.py what you will do is you will save your document inside of vector DB which will happen only one collection at a time.

And this should happen only once.

Right, so this is ingest dot PY and 2nd we can have app dot PY or you can have main dot PY. So here is where we'll have real time answer generation.

And we'll also try to create UI for this.

Which can be deployed. OK, so these are two things. One is ingest dot PY and one more is app dot PY. So whatever we discussed yesterday, right, the langsend quick start from last two to three days, I pushed the code to GitHub. It's the same URL.

What we will do today is we will use the same components. We will start off with document offline processing, then we will shunk our data and then create the embeddings and when we save it in a vector DB. So what we will try to do is we will use cloud. So yesterday what did we use?

We used in memory. Can you see this logic? So we define a path and once we define the path, it will be saved over there. Now whatever I saved yesterday, the adyantik.com, it's gone. There is no data available, right? So what we'll do is.

Ajay Patel 1:47

Mm.

Tarun Jain 1:59

Now we will save the data inside a Vector DB. So the process will be the same and then we'll try to experiment with advanced technique. Now after you define this,

what was the next step? You need to define a create collection. So when we create collection, what did we do? We define a collection.
Which has to be unique and then we have vectors configuration. Today we'll also try to use binary quantization.
Binary configuration to optimize.
Give me more usage.
And remaining steps remains the same. The only logic that will change is this part and on how you define the client and after that you have to define a vector store index which will be the same and then you add the documents. So what we will do is we will start off same on what we are repeating the only thing what we will do new today.
We will add answer generation part and we will use OPIC. So you do you guys remember what is OPIC?

Ajay Patel 3:09
I'm not for me. Opic is new for me.

Tirth 3:13
OK, for uh for uh you know, monitoring the number of tokens and.

Tarun Jain 3:13
Hello.
Yeah, so whatever LLM response is there, we'll try to monitor it. So basically this we did on Saturday, last Saturday where we were just all of us used the same LLMS and I was monitoring who wrote best prompts. So basically this is a monitoring tool.
What you guys can do is I shared the collab notebook. First we can create quadrant cloud. Can you click on this link?
Probably it will ask you for sign up.
We have completed till here retrieve context. Today what we have to do is we have to connect with LLM and then response. Since yesterday we didn't save it in cloud, all the data whatever we did it is gone. So what kind of storage is that we used in memory. Today we will use cloud and there are three ways to do it.

Ajay Patel 4:10
Mhm.

Tarun Jain 4:22
One is in memory cloud and then you have Docker. So I've also kept Docker running just to show how this works. Can you see the port number?

Tirth 4:35
Yes.

Tarun Jain 4:35
What number is 6333 for quadrant?

Tirth 4:36
Yes.
I mean.

Tarun Jain 4:46
Let me know once you're able to see this screen.
I mean even you should have this.

Ronak Makwana 4:50
Yeah, we are on it.

Tarun Jain 5:05
Is it done? You guys have logged in.

Ajay Patel 5:08
Yeah, logged in and created the.

Ronak Makwana 5:08
Yes, yes.
Stop it.

Tarun Jain 5:09
So here you can give any name. So whatever chatbot you want to build, if in case you have any PDFs that you have saved as per we discussed yesterday, either you can

have that PDF name and then you can add hyphen. So what I will do is.
I will just keep it as authentic chatbot because I want to have those four to five Web pages. So now what you can do is you can just click on create free cluster.
And once this is done, uh, within few seconds you'll have this as API key. API key will have a common format. It will starts with EY.
And this is your API key. Just copy this, come back to Colab. If not, you can also save this locally because we will be using it on VS code as well. But if it is in Colab it will be safe. So here if you see I have quadrant TPI key.
I'll just expand this. Whatever is here, I will just overwrite this and paste the new one. It's quadrant under score API under score key.
And then can you see this cull command? Inside that you have a URL. It start with HTTPS and then it will end with 6333 which is the port number. So you have to copy from HTTPS till 6333 port.
And then come back here and inside quadrant cloud you just have to paste it here. So one is quadrant API key and one more is quadrant URL.
And once this is done, you just you can just close this.
Is this done?

Ayush Makwana 7:19
Yes.

Ajay Patel 7:19
Yes, for quadrant it's done.

Tarun Jain 7:21
OK, so now what you can do is you can click on this cluster UI.
And once you click on Cluster UI, this is how your URL will look like and here you have collections. As of now there are no collections that we have created.

Ajay Patel 7:45
M.

Tarun Jain 7:45
So we can just keep this particular uh window open. I'm in this tab. I will add it here.
And this one is not required. I'll just close this.

Is this done? You add EY and then make sure your quadrant URL start with HTTPS and it ends with 6333, which is the port number.

Ajay Patel 8:02

M.

Tarun Jain 8:14

OK, so now we'll click on this Opic API key.

So you guys have to tell me what are the three things we have to use for Opic. I'll just do import OS.

Voice dot environment.

OS dot and I'll write all the three things, but the only thing is you have to tell me the keys.

Does anyone remember?

Hardip Patel 8:46

OBKPIK.

Tarun Jain 8:48

Correct one is OP KPI key then.

Hardip Patel 8:52

We have to import user data from Google for that now.

Tarun Jain 9:00

Which user data? Ha, that is fine. Hi, this one is fine. I just want these three things.

Hardip Patel 9:03

User data.

Tarun Jain 9:11

This one I'll import.

Hardip Patel 9:11

OK.
Sometime project I guess.

Tarun Jain 9:22
One is project.
OK.

Hardip Patel 9:25
Yeah.

Tarun Jain 9:28
And one more is.

Hardip Patel 9:30
User name I guess.

Tarun Jain 9:33
But what was that keyword?

Hardip Patel 9:33
Um.
OK.
Um, I guess name on Vima.

Tarun Jain 9:44
So guys click on this URL Opic API key. Once you click on this you will have to sign up either with GitHub or.
Google this thing what is called, I mean Google Gmail. So once you do that probably it will create a new screen and then you can click on this triple dots. Can you see this part? We have to click on this.

Hardip Patel 10:09
What was in a career?

Tarun Jain 10:09

And then Experiment management.
So there were three things. One is we create a new project and then we had API key and what is one more thing?

Hardip Patel 10:26
Name.

Tarun Jain 10:28
Which is opaque workspace.

Hardip Patel 10:32
OK.

Tarun Jain 10:32
Is this clear?

Ajay Patel 10:34
Can you please repeat this again for me? Actually I am just changing.

Tarun Jain 10:39
OK, OK. So I'll just repeat. So what we are trying to do now is quadrant, we understood it. We want to say what what topic we'll do is this is what we did last time.

Ajay Patel 10:42
M.
Yeah, that understood. Opic. Yeah, Opic is.

Tarun Jain 10:54
Uh, one second.

Ajay Patel 10:58
Because you know pick my account is first time I'm creating so there are no projects or anything else.

RamKrishna Bhatt 11:04
Yeah, same for me.

Hardip Patel 11:05
Project we have to create.

Tarun Jain 11:05
So for everyone, everyone I'm creating for first time. Last time what I did was I gave my API key for everyone.

Ajay Patel 11:07
Mm.
OK.

Tarun Jain 11:13
So now what we usually do is let's suppose I use any LLM and what is that LLM that we used? We used Google generate UI. What topic will do is it will track the latency of the call it took, what is the number of tokens and what is the cost.

Ajay Patel 11:18
Thanks.
Hmm.

Tarun Jain 11:29
And here you can also check what is the prompt that we used and what was the response. Now let's suppose I change the prompt. I thought, OK, this prompt is good, I got this result. Now I want to change a prompt. You will also track that part. So every time you change your prompt, if you want to see which particular response was better, you need to trace this. You have to monitor it. So in order to trace and monitor, we use OK. So this is the purpose. Now coming to the code wise, how do we usually save the data? There are three parameters.

Ajay Patel 11:52

Hmm.
OK.

Tarun Jain 12:03
One is OPIC API key. How to get the OPIC API key? If I come back here, you will see your profile icon.

Ajay Patel 12:11
Hmm.

Tarun Jain 12:12
So you click on this and here you have API key and you just have to paste this.
So this is first. Second is opaque workspace. Opaque workspace is in the sense, let's suppose I create a project inside, not project. I'm creating a space. Let's suppose I have a very big customer called.

Ajay Patel 12:20
Yeah.

Tarun Jain 12:36
XYZ OK under that XYZ I have two projects that I'm building so that XYZ is my workspace. In our case, since we used Google right by default, whatever your Google ID name is there, that is your workspace.

Ajay Patel 12:37
Hmm.
Hmm.

Tarun Jain 12:52
What is the workspace here?

Ajay Patel 12:52
OK.
They're over here for Tarun Arjun and for me.

Tarun Jain 12:56
So now I can create new workspace if I need. Let's suppose I create a new workspace called Atyantik. Under that I have two projects, one is to track rack and one more is to track agents. So I'm creating two projects now.

Ajay Patel 13:11
Hmm.

Tarun Jain 13:12
So whatever workspace is there, that is your main folder. Under that the subfolders that you have is your project.

Hardip Patel 13:19
Mhm.

Ajay Patel 13:20
OK, OK.

Tarun Jain 13:23
So here Tarun Arjun is my main thing and then under that I have different projects like Atyantik, the Neuropython, Evals, tutorial and video. These are my sub projects.

Ajay Patel 13:25
Mm.
OK.

Tarun Jain 13:33
So now what will I do? I will just create a new project.
And tell it as chat bot.
And I'll tell that's the on the website chat bot and then I'll keep it public.
Are you guys able to see this screen as well? You just have to click on new project.

Ajay Patel 13:56
Yes.

Hardip Patel 13:59
Yeah.

Tarun Jain 14:00
And once you click on your project, you should have name, you should have description and you should have project visibility.

Ajay Patel 14:12
Over here you can see there are two options like public and private, but when I am trying to create there are no options like only.

Tarun Jain 14:12
Want to see them?
OK, so you will have to click on this three dots then Experiment management.

Ajay Patel 14:22
Mhm.
OK.

Tarun Jain 14:25
Then new project.

Ajay Patel 14:28
OK.

Tarun Jain 14:31
I'll repeat this nine dots, then experiment management, then you will see the screen, the new project.

Ajay Patel 14:37
Mhm.
Hmm, public.

Hardip Patel 14:47

OK.

Tarun Jain 14:50

And then public create. So now what is my workspace? My workspace is nothing but Tarun R Jain. This might be different for you guys, so all these three variables will be different.

Ajay Patel 14:52

Hmm.

Hmm.

Hardip Patel 15:04

Um.

Tarun Jain 15:05

You but if you have used chat bot then project name is same.

Let me know if you have completed till here.

Hardip Patel 15:19

Yeah.

Ajay Patel 15:19

Just a minute now.

Tarun Jain 15:23

Oh, all of you. Anyone is still facing issues? It should be public. I hope you have selected public.

Ajay Patel 15:30

Hmm.

Hardip Patel 15:32

Yeah.

Tarun Jain 15:32
Let me delete one or two of.
OK.

Ajay Patel 15:41
I I I just want to see those code for.

Hardip Patel 15:46
What are you?

Ajay Patel 15:46
Uh, it's something.

Tarun Jain 15:53
Oh, what color?

Hardip Patel 15:53
And our project? No, I'm just happen.

Tarun Jain 15:59
And then we need Google API key.
I user data dot get Google API key. This one also I'll use user data.
Just cross check if you have used the same names, OPIC API key, then OPIC workspace. Everything is in capital, then OPIC project under score name.
Let me know if we have done till here.

Tirth 16:48
Thanks.

Tarun Jain 16:50
OK, so for data ingestion I'll be using Collab so that I can show you output at every single line. But what you guys can do is you can directly use VS code. And how do you use VS code? First thing is you have to create virtual environment.
This syntax is Python 3 hyphen M. VNV is nothing but virtual environment and then

ENV. So this ENV can be any name. Then you will do source. After source you will have ENV, then bin, then activate.
These are the first two lines that we have to create. Once you create these two things, you can create two files. One is ingest.py which is to save the data inside Vector DB and then app.py for answer generation and to create UI.

Tirth 17:39
Mm-hmm.

Tarun Jain 17:39
Once you do this then you have to install all these things. Line chain, Line Chain community. This is for text splitters.
Embeddings.
And as well as there are multiple things, prompt will come from here and even the loaders will come from here. And then this is the LLM component. And since we'll be using PDF, OK, I will not use PDF, I'm using the website, but you guys have PDF right? So what you can do is you can use by PDF form.
This is for PDF dependency.
If you are using loaders as by PDF.
And fast embedded is for embeddings.
And quadrant is for vector database.
And OPEC is for tracing and monitoring the entire pipeline.
And we can just run this. But if you're running it on VS code, don't include this exclamometry mark. You can remove this exclamometry and what you can do is you can add everything in one line only. So pip install line, same line, same community, you can add everything in one line.

Hardip Patel 19:14
And don't we have to add the quadrant variables?

Tarun Jain 19:20
Uh, that we can directly use it when we define the client.

Hardip Patel 19:24
OK.

Tarun Jain 19:26
So here it is asking me to restart. I'll do cancel because there are still pending.
So for this line it will ask me to restart, but in VS code you don't have to worry about that part.

Ajay Patel 19:42
Those Python environment creation and all those things that is applicable for when we are using AVS code, right?

Tarun Jain 19:49
Yeah.

Ajay Patel 19:51
OK.

Tarun Jain 19:52
So let's suppose land chain Google Gen. AI, the version is new now and that version is not supporting of what project you have already created. So why do we need to create this virtual environment whatever package we install?
It has certain versions if you see it. So launch important 0.2.0. Now if this is new release, let's suppose the release 0.3 and 0.3 is not working well in your current application. So during that time what you have to do is you create a virtual environment.
You install whatever packages you need. So once you create that environment, you should only use a particular package. If you want to upgrade, you can upgrade. What you can do is you can just do hyphen you and then write the particular command if you want to upgrade if you don't want to upgrade.

20:44
OK.

Tarun Jain 20:46
That environment is safe for that particular version and you don't have to worry if

any frameworks upgrade their particular version. So now if Opic upgrade their version, you don't have to worry because you are running it in a virtual environment.

Ajay Patel 20:50
OK.
Hmm.

TJ **Tarun Jain** 21:05
But if you think whatever new release has been done that is very useful to you, then what you can do is you already have Opic, you will come back to your VS code and you will just run pip install Opic but you will add hyphen you.
Opic. So this iPhone U is upgrade.

Ajay Patel 21:26
OK, and for apart from this Kolar notebook, we also need to have a VS code locally.

TJ **Tarun Jain** 21:33
Huh.

Ajay Patel 21:33
For this exercise.

TJ **Tarun Jain** 21:42
So VS code everything will be added here in requirements dot TXT file.

Ajay Patel 21:45
Yeah, yeah, that understood requirement of TXT.

TJ **Tarun Jain** 21:50
So once you do this, what you can do is there is a command called pip freeze.
Let me know how many of you are trying VS code. Can I know the number? How many of you are trying on VS code?
OK, so.

Tirth 22:05

I'm on.

TJ **Tarun Jain** 22:07

What you can do is it after you create virtual environment and after the installation is done.

You just have to create a file called requirements dot TXT.

Tirth 22:20

Uh, with the free time done that.

TJ **Tarun Jain** 22:22

OK, then fine.

Ajay Patel 22:24

Thanks.

TJ **Tarun Jain** 22:25

And once this is done, you just have to do.

Pre and then requirements dot EXT. So whatever package is there in your current virtual environment that will be appended in this particular file.

Ajay Patel 22:44

OK.

Nope.

OK, so the ingest dot PY. Before creating this ingest dot PY and app dot PY or main dot PY in locally VS code, we need to do the exercise for setting a Python environment and for if we are using a collab research then we don't need to do this.

TJ **Tarun Jain** 23:02

Correct.

Uh, these six steps are not required.

Ajay Patel 23:08
OK, OK, OK.

Tarun Jain 23:09
So this this six steps is only for those who are working on VS code because we will eventually move to VS code when we do answer generation because I need UIUI. I can't do it on Pollab.

Ajay Patel 23:15
OK, OK.
OK.
Pull up.

Tarun Jain 23:27
But the code is same. I'll just copy paste the same code what we are doing in Colab.

Ajay Patel 23:29
Mm-hmm.

Tarun Jain 23:34
The only thing is like if you use collab right, most of the time there are some piece of code you don't want to rerun. In VS code we have to rerun everything. If there is any LLM component which I don't want to use any LLM.

Ajay Patel 23:45
Yeah.

Tarun Jain 23:51
Obviously you can comment, but sometimes we just rerun everything.
That is, collab is just for safety purpose, that's it.
Till here is done, the installation is done. I will do runtime, then restart session. I'll click on this and then again I will rerun this particular part.

Ajay Patel 24:03
Yeah, yes, yes.

Tarun Jain 24:13
It will ask me to grant taxes. I will grant taxes.
OK, So what do you think is the first step?
If I type probably it will do auto type because in this sub heading.

Ajay Patel 24:34
For the ingestion, so we'll need.
Grapple.

Tarun Jain 24:40
What is the first step?

Ajay Patel 24:43
A scrapper. Uh, what we got?

Tarun Jain 24:44
So I'll write both the lines of code if you're using PDF as your data, so from.
Lunching community dot document loaders.
Import I'll do web-based loader comma by PDF from 2 loader.
And what is the second line?
After we have the data, what are we supposed to do?

Ajay Patel 25:26
Oh.
OK.
Reading the vector.

Tarun Jain 25:34
No, what is the second shot?

Ajay Patel 25:34
Chunks chunks what?

Tarun Jain 25:37
We have to convert it into chunks from line chain.
Community dot OK, it's text splitters actually.
Import recursive.
Character text splitters then.
This line you should tell me.

Ajay Patel 26:03
Which one? Convert to OK.

Tarun Jain 26:08
So document loaders is done, splitting is done. Next two to three lines I will not read.

Ajay Patel 26:12
Hmm.

Tarun Jain 26:18
So what are the next three lines?

Ajay Patel 26:22
Create AI mean those chunks should be stored in a vector, created in a embedding.

Tarun Jain 26:27
I'm ready so.
What embeddings are we using?

Ajay Patel 26:33
Gina.

Tirth 26:35
Tina with fast and then.

Tarun Jain 26:35
So what is the logic for that? Here we are just importing.
So from.
We discussed this yesterday. How do you define embeddings?
You can check the yesterday's code.

Hardip Patel 27:06
Langen under score community.

Ajay Patel 27:07
Pradeep, you're on mute.

Tarun Jain 27:09
Lunch in community then.

Hardip Patel 27:13
Dart embeddings.

Tarun Jain 27:15
Embedings.

Hardip Patel 27:16
Input.
Oh, sorry, yeah.

Tarun Jain 27:19
OK.

Hardip Patel 27:22
dot dot Gina.

Tarun Jain 27:25
Hi, if you're using Xena, we can use Xena, but here we are using fast number.

Hardip Patel 27:28
Yeah, OK, then import faster embed embeddings.

Tarun Jain 27:31
What?
OK.
Cool. Now last two lines are pending.

Hardip Patel 27:40
M.

RamKrishna Bhatt 27:42
We have to define model right? I think in fast time we'd.

Ajay Patel 27:45
No, no, no.

Tarun Jain 27:46
Huh. So.
So that we will define next like we'll define embeddings.
And then we will define that. But here we just have to import, so there are two more import statements left.

Tirth 27:59
Line field about rank conference vector score.

Tarun Jain 28:02
Uh, so from lengths in quadrant.
I just have to import quadrant.
Vector store and then.

Tirth 28:14
And.

Ajay Patel 28:16
Quadrant client, quadrant client, quadrant. Yes, quadrant.

Tarun Jain 28:19
Oden climb.
Blind import.
Quadrant client and there was one more line we used that was from quadrant client HTTP models import these things.

Ajay Patel 28:36
Distance on vector patterns.

Tarun Jain 28:37
This is for cosine similarity and this is for defining both of them which is size will be 768 and distance dot cosine that we need to define inside vector configuration.
You can also add more.
This is clear.

Ajay Patel 29:00
Hmm.

Tarun Jain 29:01
So we did just this part, whichever is covered inside this red. So external data. Once you have external data, it can be anything. If you guys want to build a chat bot for any YouTube video, you can pick a YouTube video as well and then you have chunk data and then you have embeddings and then.
You have to define a Vector DB. So Vector DB. First you need to create a space and then create a collection. Once that collection is created then you have to append the data. So for appending the data we'll be using quadrant vector store.
So these are the import statements.
OK, so now what we can do is I'll define URLs.

Hardip Patel 29:47
So.

Tarun Jain 29:51

You guys can just upload your PDF.
And once you upload that PDF, just define a path.
Path equals to content and that particular PDF file. So you just have to click on whatever you uploaded. Double click, rename. Sorry, not rename.
Copy path and once you copy path you just have to paste that here.
So in your case it will be content then.
The file name that you had dot PDF.
Here let me just copy paste those five web pages that I had.
Heard.
And what is the next step?
So here we just have two lines of code. The first line of code will be loader. So whatever loader you want to define here I'll be using web base and inside web base what is my variable? It is web path. So what I'll do is I'll just pass this URL here.

Ajay Patel 30:47

OK.
It's my name.

Tarun Jain 31:04

And then next step is I need to get my documents documents equals to loader dot load.
So here we completed parsing of data, which means I uploaded a source of the data that I have and then I'm just extracting the raw data.
So we completed this part now with the code which is our external data.
Till here everyone is done.

Ajay Patel 31:44

OK.

RamKrishna Bhatt 31:47

I'm not yet working with it.

Tarun Jain 31:50
OK, I guess uh if the file is big it will take some time.

Hardip Patel 31:51
Yeah.

Tarun Jain 31:58
How many of you are using PDF? You can just raise.

Hardip Patel 31:59
It took. It took me two seconds. I'm using PDF. It took me just two seconds. Is it alright or not?

Tarun Jain 32:06
No, that's fine.
Can you do length of dogs?

Hardip Patel 32:12
Yeah.

Tarun Jain 32:12
So after this you can just check the length of.
Documents.

Hardip Patel 32:21
357.

Tarun Jain 32:23
What, 357?

Hardip Patel 32:25
Yeah, I think, yeah, that that was the amount. It is career level thing guide.

Tarun Jain 32:33
OK, how many pages does it has? 300.


Hardip Patel 32:35
Yeah, 3300 about 300 case.

Tarun Jain 32:39
OK.
Uh, is it? Is this done?
So since I have only four data, what I will do is I'll also create chunks. So for that I'll first define splitters.


Ayush Makwana 33:02
Yes.

Tarun Jain 33:10
Which is recursive character text splitter and I need to define chunk size. So chunk size I'll keep it 2048 and my chunk overlap I'll keep it 0.
And now why am I using split documents? Because the format of loader if I do documents of 0th index.
It is in document format. So what do I need to do? I just have to use docs equals to splitter dot split documents. I will just pass this variable. This is the same thing. I'm not writing any new code from yesterday. We first use the source files.


Hardip Patel 33:49
Oh.

Tarun Jain 33:49
Then define the loaders and then split it.
And here we can define Zena if I come back.
Supported models.
Copy and the variable is model name. If you use model you will have a default base model and the base model dimension is 384.
And what will this line do now? It will download the particular model.

Are you guys following till here? It's the same thing. We haven't written any new code yet.

Hardip Patel 34:48
Yeah.

Ishan Chavda 34:49
Good.

Tarun Jain 34:55
OK, I'll wait for one minute. Let me know if you have downloaded the particular model or not.
I'll just remove this part so that I can fit the particular code selling on the screen.
So since I'm using Gina, this will be 368. So how do I test it? I can just do length of.
Embeddings dot embed query. I'll just ask a simple question. Hello world.
And it should show 768.
Is it done?

Margi Varmora 36:21
Yes.

Tarun Jain 36:22
All of you.

Ishan Chavda 36:27
Yeah.

Tarun Jain 36:28
OK, So what is the next step?
So we completed extracting of data. Now we also have chunks and then what am I trying to do? I also created the embeddings. Now what is the next step?
OK.

Ajay Patel 36:49

We need to store it in a quarter database. Previously we stored in in memory, so now we need to store it in quarter database.

Tarun Jain 36:51
OK.
So I'll use this quadrant client. So for quadrant client what do I need? I need one URL and then I need API key.


Ajay Patel 36:58
OK.

Tarun Jain 37:05
If you have saved it here, what we can do is we can just do user data dot get whatever name you have defined. Here it is quadrant URL. I'll define quadrant URL then user data dot get.
I'm defining it in a variable.
And then I just have to define the client client equals to.
Modern client here if you notice yesterday we used path. Apart from path you have location which is mainly used for local where you have to give your local host and 6003 33 which is running on Docker.
If not, you have to use this URL which we have already defined and apart from URL what is the another thing? We have API key. So if I print this URL.
If you notice you will see a region. So the region is Europe W 3 and then you will see the cloud provider which is GCP cloud. So similarly if you are using AWS or if you are using Azure you will have different name here and you might have seen a different location. So this is what is your endpoint URL.


Ishan Chavda 38:21
Yeah.

Tarun Jain 38:25
And you just have to give URL equals to URL then comma API key equals to API key.
And then you can define any correction name.
Collection name I just tell website chat bot.

Or are you just in web pages?
Let me know till here if it is done.

Margi Varmora 39:07
Yes.

TJ

Tarun Jain 39:11
OK, so now what we will do is.

RamKrishna Bhatt 39:12
Um, one question. Sorry to interrupt. I am unable to import length and quadrant.

TJ

Tarun Jain 39:14
Yeah.

Ajay Patel 39:18
You have you run in.

TJ

Tarun Jain 39:21
Have you installed this part?

RamKrishna Bhatt 39:23
I think, uh, I missed that. Yes, let me do that.

Ajay Patel 39:30
Link in quadrant. It should be there.

RamKrishna Bhatt 39:32
Yeah.

TJ

Tarun Jain 39:38
Opic is also there where we depend Opic.

Hardip Patel 39:38

Find me.
For me, fast time bed is taking insane enough time.

Tarun Jain 39:46
What?

Hardip Patel 39:47
Like uh, it is still loading. Uh, fast time bed. I don't know.

Tarun Jain 39:51
Fast and baby.
Uh, can you see what is the time here? Is it connected or is it disconnected? Usually it should not take that much long.

Hardip Patel 40:01
OK.
Where time OK.

Tarun Jain 40:05
Hi here here we'll see seconds actually like if there is any.

Hardip Patel 40:08
Yeah, yeah, it it no, it shows in the exclamatory.

Tarun Jain 40:17
Examatry. I guess there is some error then.

Hardip Patel 40:21
OK.

Tarun Jain 40:25
Can you just close it correct?

Hardip Patel 40:25

Uh, can I?
I guess I can refresh, right?

Tarun Jain 40:30
Oh yeah, you can.

Hardip Patel 40:33
All right, I guess I'll wait.

Tarun Jain 40:36
Follow till collection name.
OK, so now what we will do is what is the syntax to create collection? Client is already defined. I'll just copy this client variable. Inside this client variable we have create collection.

Hardip Patel 40:43
Um.

Tarun Jain 40:56
And inside create collection the first parameter is collection name. So whatever auto complete as given that is correct. One is you have collection name equals to collection name. Then you have vector configuration equals to models vector params size 768.
And distance equals to distance dot cosine. Now what I need to do is I also need to add quantization config.
So you have this quantization conflict. Inside this we need to define binary quantization. So for that I will have to import it from models dot binary.
Right.
Binary quantization. Yeah, the first one.
Are you following? So we use we usually we usually define collection by client dot collection name. The first parameter is collection name. Then you have vector config which is for dense vectors.
So this is for dense vectors.
And the size of the fixed dimension is 768 and the distance measurement is cosine.
And then we have quantization conflicts, right? We want to say memory.

Hardip Patel 42:05
Um.
Mm.

Tarun Jain 42:17
We want to save memory.
So that's the reason why we are using quantization conflict equals to models dot binary quantization. Inside that there is one variable called binary. If you see here you have binary. Inside binary you have binary quantization conflict.
So I'll just define binary equals to binary quantization. This is the one quantization conflict. Here I just have to keep always RAM to be false.
So what will happen if I keep always RAM to be true? It will hit my RAM limit.
So I don't want to hit the RAM limit.
Let's just confirm if you have written this correctly.
So where did I define these models? I've defined these models.
From quadrant client import quadrant client comma models this line.
Oh.

Hardip Patel 43:38
Right.

Tarun Jain 43:45
So tomorrow's session what we'll do is after you define create collection, instead of using quadrant vector store from line chain, we will use quadrant directly.
Instead of using length chain.
We will use quadrant functions directly.
Why? Because I need to have my own metadata. Own metadata. Where is metadata added? What is the keyword?
Does anyone recall the keyword?
Where do I save?
The metadata if I'm using vector database.
Do you guys recall where do we save metadata?

Ajay Patel 44:56
Not sure, not sure for me.

Tarun Jain 44:58
OK, so I'll open this notebook again.
So here if you see under vector database I wrote two points whatever page content we have. So what is page content? If I just do documents of 0 dot there are only two variables, one is page content and one more is metadata.
Right. So whatever page content you have that is vectorized and then whatever metadata you have, it is passed in the payload, right? So if you want to add your own custom metadata to payload, what we will do is instead of using land chain.
Which is your quadrant vector store. We will directly use quadrant. So the functions are a bit different here. If you see you actually defined your quadrant vector store and then you used a parameter called add documents. So basically what is happening is.
The logic of add documents is returned by Land Chain. So this logic is there right? What is this? This is the class which is from Land Chain and this is the method that is inside quadrant vector store. So what we will do is what function is used inside this add documents. We will pick that function from.
The given document and we will directly use that and make modifications. Is this clear? Instead of using lines, we will directly use what are the functions that is used within this method.

Ajay Patel 46:28
Portland.
OK.

Tarun Jain 46:36
So this is what we'll try to do tomorrow, but till we are, have you guys understood what we are trying to do?

Ajay Patel 46:38
Mhm.
Yeah.

Tarun Jain 46:43

So we create an empty space and then in that empty space we are defining web pages. And what is the configuration of web pages? Whatever embedding dimensions you have, which is 768, you need to define that and then the distance is cosine similarity. Why? Because we are using dense.

If you are using sparse, it is TFIDF.

And BM 25.

So anyone has any doubts? Because this is very important. Till here whatever you have done, you will repeat this process.

Right. This process is same. You start with loaders, extract raw data, split, then embeddings. Whatever embedding size is there, you have to calculate that so that you can give it to your dimensions. Here there are three ways cloud.

Memory and local. For local we'll use Docker and then you create a space and save it here. And now I now if I run this it should be true.

Till here is it done? I hope you understood the process. This this is the same process we will continue for every single drag project.

Let me know once you get through.

Ajay Patel 48:08

OK, just a second. I'm like to name.

Hardip Patel 48:11

Yes.

Ajay Patel 48:15

Chosen vector config models.

This models we need to import now this because I've got.

Tarun Jain 48:27

So where we have defined this quadrant client rate quadrant client comma models.

Ajay Patel 48:36

OK, yeah, OK, OK. That one was missing part for me.

Tarun Jain 48:37
2.

Ajay Patel 48:41
Modern client.
Moments.

Tarun Jain 48:45
So from quadrant client import quadrant client models.

Ajay Patel 48:50
Now it's true.

Tarun Jain 48:53
It's true for everyone. So now I told you to open this tab, right? You can just refresh this.

Hardip Patel 48:56
Yeah.

Ajay Patel 48:56
Yeah, yes.

Tarun Jain 49:01
You should see web pages.

Ronak Makwana 49:04
AUI partner.

Tarun Jain 49:04
And what is the values? The size is 768, the distance is cosine.
As of now, there is no data inside it because we didn't add any data yet.

Hardip Patel 49:18
M.

Ajay Patel 49:23
Hmm.

Tarun Jain 49:25
Can you see your uh collection name whatever you have defined? Cool.

Ajay Patel 49:26
Yes, yes, yes.

Tarun Jain 49:31
Then what is the next step? Now this step will take too much of time because all of us are using actual data, so we have to define vector store.
Equal show.
Vector.
It's quadrant vector store. What are the three parameters collection name?
Equals to collection name.
And then we also have client equals to client and what is the last parameter?
It is embedding without S embedding equals to embeddings.
And here I just have to use vector store.
dot add documents.
And I just post chunks.
This will take time.
OK, used variable for docs.

Hardip Patel 50:49
Yes.

Tarun Jain 50:56
So what is this process called?

Ajay Patel 51:02

Uh, storing those uh.
Storing this vector M.

Tarun Jain 51:06
But what is the terminology for this?
So there are some terminologies we'll have to keep in mind. One is vectorization.

Ajay Patel 51:21
Mhm.

Hardip Patel 51:22
Condition.

Tarun Jain 51:24
Which is for page content.

Hardip Patel 51:25
Yes.

Tarun Jain 51:28
And 2nd is payload. Why is payload used for metadata or you can say additional data. So I'll tell one use case right for payload. Let's suppose you're building an application for a food industry, right? So.

Hardip Patel 51:35
So.

Tarun Jain 51:44
In food industry you have your own documents that you have saved in PDFs. Now what we need to do is if anyone asks what is the best food, that's it. It will just ask a simple query. What is the best food?
Near me. If the question is like this and what will your vector database do? It might rank some documents which will be from Bangalore, it will be from Delhi and it will be from Mumbai. But if you are best food near me, if it is Gujarat.
The.

Hardip Patel 52:17
Tarun, just as like I guess like my I have like your session crashed after using all available then. Should I start with default or something?

Ronak Makwana 52:25
With.

Tarun Jain 52:30
Oh, what happened?

Hardip Patel 52:33
Uh, it is showing me that your session crashed after using all available RAM.

Tarun Jain 52:40
Oh, yeah.

Hardip Patel 52:40
When adding documents.

Tirth 52:40
Use VS code.

Hardip Patel 52:44
OK.

Tarun Jain 52:48
No, it should not hit it. But yeah, what you can do is you can click on runtime, then click on change runtime type, then select D4 GPU.

Hardip Patel 52:50
That's why.
Yeah, it's OK. OK. Sorry, please.

Tarun Jain 52:59

OK.
OK, so here there is one use case. Let's suppose metadata we added, but what about the additional parameters? So the use case what I was talking about is you ask a query called what is the best food near me, but you are in Gujarat and you're not specifying this particular part.
Now what vector database is doing is it is giving you the results from Bangalore, Delhi and Mumbai. So technically this is a false response so you can extract details from geopolitical location as well geopolitical.

Hardip Patel 53:27
So.

TJ **Tarun Jain** 53:35
Geopolitical location. So this is again one parameter that is defined in payload. So payload mainly defines how you want to filter your filter your retrieve documents.
Is this clear?

Ajay Patel 53:54
Yes.

TJ **Tarun Jain** 53:54
And last is you have indexing. Now what is indexing? It's nothing but ingestion.

Hardip Patel 53:55
Yes.

Ajay Patel 54:01
Hmm.

TJ **Tarun Jain** 54:02
So ingestion is a process where you.
Have your own data, you create the embeddings of it and save it in a vector DB. So that process is called indexing or ingestion, which we currently did just now.
Anyone has any doubt in this three terminologies?
So if you open any documentation, right, not just quadrant, any documentation, you

will see those concept. So vectors I mentioned. Then you had payload search is in search in the sense once you save your database, right? Once you save it inside Vector DB, how do you search?

Ajay Patel 54:34
Mhm.

TJ

Tarun Jain 54:41
So you had similarity search.
Then you had similarity surge.
Based on scores, which is thresholding basically. And then you had MMR which is maximum marginal relevancy. These are some of the search techniques, right? And then what's next?
Payload I already said and then indexing. Indexing is nothing but the same. You save your data set inside a vector DB and then you have optimizers. Optimizer is where we have used binary quantization.
Is it clear this main terminologies?

Ajay Patel 55:23
Yes.

Hardip Patel 55:23
Yes.

Ronak Makwana 55:24
Yeah.

TJ

Tarun Jain 55:24
Vector payload, indexing and search.
Let me know if in case your data is saved. How do you get to know? Come back to your.

Ajay Patel 55:37
UA.

Tarun Jain 55:38
Dashboard refresh this you will see this number. So this number is equivalent to your chunks. So if I do length of docs.

Hardip Patel 55:47
OK.
Oh, thank you.

Tarun Jain 55:51
It is 45. This is also 45. Now what is this?

Ronak Makwana 55:51
Punch.

Hardip Patel 55:56
Default also use that line.

Tarun Jain 56:01
What even T4 it is hitting?

Hardip Patel 56:04
Yeah.

Ronak Makwana 56:04
Yeah.

Tarun Jain 56:05
Uh, then we can do one thing.

Hardip Patel 56:08
That's fine. I guess I will do it later.

Tarun Jain 56:11
Oh, one second. I'll just tell you one approach on how you can avoid that. There is a

parameter called on disk.
Restart on.
Huh. On disk payload, keep this as false.

Hardip Patel 56:35
OK.

Tarun Jain 56:37
This should will be true on disk in the sense you need it to run on disk and not on run.

Hardip Patel 56:41
OK.

Ronak Makwana 56:42
OK.

Tarun Jain 56:42
So if you keep it as false, it is running on RAM. If you keep it as true, it is running on disk and disk space is there a lot.

Ajay Patel 56:51
Mhm.

Ronak Makwana 56:51
OK.

Hardip Patel 56:51
OK.

Tarun Jain 56:53
But do one thing, don't do restart session. Click on disconnect runtime. Whatever RAM is there, make it zero first and then run it.

Ronak Makwana 56:57
Mm.

Hardip Patel 56:59
Mm.

Ronak Makwana 56:59
M.

Tarun Jain 57:04
And before this what you can do is just do import GC.

Hardip Patel 57:05
OK.
Mhm.

Tarun Jain 57:11
And what will you do? What variables we don't need? We don't need documents now because why I'm creating the chunks of it. Whichever variable do you think it's not required, just use import GC and do GC dot collect.

Ajay Patel 57:11
Garbage collector.

Hardip Patel 57:15
Mhm.
M.

Ronak Makwana 57:27
Hello.

Hardip Patel 57:27
Mm.

Tarun Jain 57:27
Here I'll do delete documents and then I'll just create.
So this you have to run before you create collection and you can add this variable.

Ronak Makwana 57:39
OK.

Hardip Patel 57:40
OK.

Tarun Jain 57:41
Only to those who who are eating the Ram.

Hardip Patel 57:48
OK, I will. You said like I can restart, right? Delete runtime and restart.

Tarun Jain 57:51
Not restart, disconnect, disconnect and delete and then rerun.

Hardip Patel 57:55
Oh, OK.

Tarun Jain 57:58
Can I confirm how many of you have completed till here?
This is same what we did yesterday.

Ajay Patel 58:04
Yes, till that I guess that's complete.

Tarun Jain 58:08
OK, so now what we can do is either we can run code here, if not we can come to VS code and create a new file. So whatever you do from now on, once your data is saved, you don't have to use your document again.
So your use of document is gone. We don't have to use this again. So the only

terminology that we are supposed to use again is embeddings. Let's suppose you're creating it from scratch. You're creating app dot PY from scratch. What will you do? You will start with embeddings.
Which is your same embeddings. I'm just rerunning it.
And then what will you do? You will again define your client.
Assuming your URL and API key is already defined.
You understood what I'm trying to do. Even though these variables are already defined, I'm repeating this here because usually this is in a different file. Now this is your app dot PY.
Whatever we did till here, it is your ingest dot PY.
Oh, is this clear, right? Because you're running it on.

Ajay Patel 59:21
Yeah, yeah, yeah.
Yeah.

TJ

Tarun Jain 59:24
So after embeddings we have client and after client.

Tirth 59:28
So I moved the above to ingest dot PY and now I'm working with main dot PY.

TJ

Tarun Jain 59:38
May not be OK.
So now collection name. After this collection name you just have to define your vector store. So instead of vector store, what I'll do is I'll define it as DB just to make it different. Now whatever you have with here, it already have your data saved. So I'm not using vector store, I'm using DB just to have different variables.
Here the data is already present.
And what will I do now? First I will test if my vector database is working or not. Then only I will proceed with the LLM part because I don't want to unnecessary waste the LLM tokens.
Is this clear?

Ajay Patel 1:00:22
Yep.

Hardip Patel 1:00:23
Yeah.

Tarun Jain 1:00:24
So same thing I'm repeating what you did in ingest dot PY because if you're doing it on the influence level, you have to define embedding again, same embeddings that was used to save your collection name, right? So what was the embeddings used? The fast embed?
I'm defining the client and now this DB, whatever I'm defining already has the data. Initially it didn't had the data, so we added the documents. Now it already has it so. Best.
Docs equals to DB dot. I'll use maximum marginal relevancy and what is the input that I need to pass here?
What are the two input variables? One is user query.

Hardip Patel 1:01:12
OK.

Tarun Jain 1:01:13
I just tell how do I contact?
So what is the second variable?

Tirth 1:01:25
A equal to 4.

Ajay Patel 1:01:27
K equal to seven.

Tarun Jain 1:01:29
I'll keep it as three or four. I'll just keep it 3.

Hardip Patel 1:01:30

OK.

TJ **Tarun Jain** 1:01:35

Because every single token is 2000. So now how much tokens am I passing? Roughly around 6000.

Or less than it also. So not every time it will be exactly 2048. Maximum is 2048.

Tirth 1:01:52

M.

TJ **Tarun Jain** 1:01:53

So this is my upper limit 6000.

So I'll just define user query.

So query equals to user query comma.

Ajay Patel 1:02:08

Right.

TJ **Tarun Jain** 1:02:12

K equals to K.

So if you notice I'm using DB, I'm not using vector store.

Ajay Patel 1:02:23

Mm.

TJ **Tarun Jain** 1:02:25

Now test docs. If I do the length of this it should be 3.

And here I just have to use dot page content.

OK, the 0 index.

And then first index.

Now can you tell me how filtering technique will be useful here? Just make an assumption, where do you think the filtering technique will be useful? Because as I just found where I can give you that example.

Hardip Patel 1:02:58
Mhm.

Tarun Jain 1:03:17
So filtering in the sense, what do you think is the metadata for this thing? What is the metadata for docs?

Tirth 1:03:27
And then you are in.

Tarun Jain 1:03:30
So you have URL, you have title, description and then language. So with just this 4 variables, where do you think we can apply filtering?

Tirth 1:03:50
And give URN filtering with the content or the type.

Tarun Jain 1:03:56
Let me show you some difference. So if you notice, I asked a question called how do I contact Advantic? So this is a very simple prompt. OK, so there might be high chances whatever prompt you add here might be from a different web page. So basically typically what I was expecting when I asked.
How do I contact Atyantik? The first source should be about page instead of this as development.
So you understood what I'm getting at. So I was expecting at yandik.com but the source that I got is SARS development. So now what I will do is when I do filtering technique I will tell hey whatever prompt I'm asking it should come from.

Hardip Patel 1:04:23
Mm.

Tarun Jain 1:04:39
This particular source and that source is homepage.
You got the problem statement that I said.

Tirth 1:04:49
And.

Tarun Jain 1:04:50
So if I ask any other question, let's suppose I ask XYZ. Now this XYZ is from coming two documents. One is from SARS development and one more it is from about us. So now what you can do is when you create a UI you can say hey I'm asking this question.

Hardip Patel 1:04:51
Yes.

Tarun Jain 1:05:10
And I believe you should response this from this particular web page. Either you can let LLM to decide this or what you can do is you can add add a filtering layer. So that filtering layer is I want XYZ to be answered from about us.
Not from SARS. Now what will happen is when you print the K equals to three, you will not see any SARS. You will only see about us in all your source in all the three source. So that is where filtering is important and for filtering you need metadata.
Is this clear?

Hardip Patel 1:05:51
Yes.

Tarun Jain 1:06:03
OK, let me know if you're able to print the response.

Tirth 1:06:09
Yeah, you want to do that.

Tarun Jain 1:06:11
OK, so this is now happening on app dot PY, not on ingest dot PY.
OK, so now what we have to do is we start with first process. What is the first one retrieval which we already did?

So retrieval is nothing but you have test docs, then you are adding DB dot search. So what is search doing? It is retrieving the relevant document so that is your first R. What is the A? What do we do in A?

Hardip Patel 1:06:45
Yeah.

Tirth 1:06:48
OK.

Tarun Jain 1:06:57
So we are just left with this two-part. Now R is done, which is a retrieval. What is A&G? What do we do in here?

Ajay Patel 1:07:05
Use an LM bug.

Hardip Patel 1:07:07
And.

Tirth 1:07:07
London.

Tarun Jain 1:07:08
No, he need.

Ishan Chavda 1:07:11
I think we will connect with her element.

Tarun Jain 1:07:15
A what?

Ishan Chavda 1:07:19
I think we need to think with.

Tarun Jain 1:07:22
We have to use the.


Ishan Chavda 1:07:25
Think with Adele.

Tarun Jain 1:07:25
In operation, what happens?
So what is LLM use for? LLM is like you give a prompt, it generates.


Ishan Chavda 1:07:34
OK.

Tarun Jain 1:07:35
So LLM stands for generation, right? So it will come in G So what will happen in augmentation?


Ishan Chavda 1:07:44
Uh, maybe.


Tirth 1:07:45
So you know there are three steps. First retrieve the document documentation. We take the user query and the retrieve documents again to create the right context.

Tarun Jain 1:07:59
So that is our human prompt. So what happens in augmentation? We are defining a. Prompt. Why is it?
So basically augmentation is I have to define a template, template of a prompt and the logic of the prompt is I need a system prompt under.
Human prompt.
So what will my system problem look like?
So what do you think is the system prompt here?

Tirth 1:08:52
Export Data Expector.

Tarun Jain 1:08:56
No. So what is my data it's related to?

Ajay Patel 1:08:59
Attending website.

Tarun Jain 1:09:00
Uh, so I'll tell you are an seasoned.
Employee at Atlantic, who knows?

Ajay Patel 1:09:11
Thank you.

Tarun Jain 1:09:15
What product?
Atyantik is building.
Or what do I give here?
Dominic experts, I need some context here. You are a senior employer at.

Hardip Patel 1:09:37
OK.

Tarun Jain 1:09:37
Onos.

Hardip Patel 1:09:39
OK.

Tarun Jain 1:09:43
Shall I just use this about the software and?

Ajay Patel 1:09:46
Technical.

Tarun Jain 1:09:50
Is this fine?

Hardip Patel 1:09:54
Yes.

Tarun Jain 1:10:00
So this is role play. OK and now what it will do is I'll add some more thing. You should be respectful.
And rigsful while answering the user questions. Now here I'll add condition the only source of.
Permission you have is the context provided.
If the user query is not from the context.
Just say.
I don't know.
Not enough information provided.
So this is the system prompt user prompt. How many input variables will I have?
So how many input variables do I have for user prompt?
So we completed the red part. Now we are at this open area. So query I have added which is how do I contact and then automatically embedding model is used whenever I'm using the vector DB.

Ajay Patel 1:11:23
Mm.

Tarun Jain 1:11:33
Just one second.
So as soon as I ask the query, what is happening? I'm using it in a Vector DB. Vector DB is returning the context. So till here is your R which is your retrieval. Now what are you doing in augmentation? You're trying to create a template where you have query and context.

So now you're at the augmentation step, which is this line, the curve line on how you want to combine this query and context before you give to LLM. So how many input variables I have?

Ajay Patel 1:12:07

2.

Tarun Jain 1:12:08

Just to.

So I will just define context equals to context.

And I'll just add some placeholders here.

Or what is the best thing I can define?

XML.

Context and then I have query.

User query equals to query.

Answer.

And so.

Good portion.

Answer the question based on the context provided. So this will be capital. If the question cannot be answered using the information provided, answer with.

I don't know. Not enough information provided.

So instead of portion it will be user query.

Is this clear?

I can just copy this down. If not, I'll let me paste it. This is the system prompt.

And this is the answer prompt.

So now what next?

What's next?

If I have system prompt and user prompt, how do I convert this into a template?

Ajay Patel 1:14:03

Wrong template.

Tirth 1:14:03

On templated or a chat from template whichever we want to leave.

Tarun Jain 1:14:07
From lancen code dot prompts import.
So let me copy paste from yesterday's notebook.
So from length and core dot prompts import chat prompt template.
And then these two.
So here partial variables is not there. I have system prompt and for user AI user prompt.
And this will come inside messages.
Till here is it done? User start to add system prompt, user prompt and create a template. This is augmentation.

Ajay Patel 1:15:03
Um.

Tarun Jain 1:15:11
So retriever is done, augmentation is done and what is left generation.
So for generation we can do it in two approaches. One we can use LCL or we can use land graph.
And what is preferred Langraf?

Ajay Patel 1:15:28
OK.

Tarun Jain 1:15:30
Because this was latest released. Previously we had LCL. Before LCL you had chains. Which was completely crap. So they just remote change and they added LCL and but now since their main focus is Langraf, they're trying to create state management in Langraf. So when I said state management.
What topic comes to your mind?
We have covered state management.
Do you remember class and inside class if I define any self dot?
Uh, let's suppose.
Balance.
Can I reuse the self dot balance increment and decrement within the class methods?

So let's suppose withdraw initially have 10,000 here if I'm doing self dot balance.
Minus equals to 500 and then serve dot deposit.
Self dot balance plus equals to 500.
And now if I call this outside, when I create the object, let's suppose I create B equals to bank. Here if I change any value, is the balance also changing? So what are the attributes?
Attributes are persisted over state.
Do we recall this concept?

Tirth 1:17:26
Yes, yes.

TJ **Tarun Jain** 1:17:27
So what we are trying to do is self dot balance. If I want to change within any methods or once I create object if I want to change I can change it. Why? Because the attributes of a class variables are persisted over state. So this logic is also applied in Langraff.

Tirth 1:17:27
Yep.

TJ **Tarun Jain** 1:17:48
So what we can do in Langraf in order to create the chain, right? So here mainly you have state management.
Which is one of the major feature of plan graph and they're using graph based.
Let me not tell here if you have completed so that we can start with the generation part, which is the final step.
Is this done?

Tirth 1:18:17
Assistant.

TJ **Tarun Jain** 1:18:18
OK, so now what are we supposed to do here from?

Ayush Makwana 1:18:19
Yes.

Ishan Chavda 1:18:19
OK.

Tarun Jain 1:18:23
Google Gen. AI import.
Chat Google generative UILLM.
Here I just have to have to define model.
Was it model or model name?
I'll just copy this. So what will be the temperature? What should be the temperature here?

Tirth 1:19:01
We need accurate information.

Ajay Patel 1:19:01
Normally it's .7.

Tarun Jain 1:19:04
So it should be very close to zero. I'll keep it 0.1.
If possible, I don't know if they support 0.0.
Let me test with flash.
LLM dot invoke.
Testing. Hello world.
Yeah, so we can use 0.0.
Pillar is it done?

Ajay Patel 1:19:55
Search.

Tirth 1:19:56
But.

Tarun Jain 1:19:59
Now what is left? Opic is left. We haven't still configured Opic, right?
Hello.


Ajay Patel 1:20:09
Yeah, yeah, we understand.

Tarun Jain 1:20:10
So now what we need to do is we need to configure OPIC. So let me just see what is the documentation of it.
LLM experiment. This is what we did last time.
OK.
We have Opic API key, Opic workspace, Opic project name and from here I just have to use tracer.
So opic dot integrations dot linechain. I need to define opic tracer.
So does anyone recall where we are at supposed to adopic tracer?


Tirth 1:21:01
We added it inside the the process of all.

1:21:02
Yes.

Tarun Jain 1:21:06
Correct. So when I do invoke inside invoke there is a function called callbacks. So inside callbacks I have to define Opic tracer.
So I'll define opic tracer.
And when I define LLM, we are supposed to define.
Callbacks as Opic Tracer.
It.
Just rerun the LLM again because we forgot to add Opic tracer.
But.
We just have four or five more lines of code. That's it.
This we have repeated from Langchain Google's in AI to define LLM and since we

want to trace the entire pipeline, what are we trying to do? So for LLM we have a function called callbacks in Langchain. Inside callbacks we just have to define Opic tracer.
And this is same code that we wrote during the LLM experimentation when few people were using Samba Nova and few people were using Google's generative AI.
Let me know till here if it is done.

Ajay Patel 1:22:39
Yeah, it's done for me.

Tarun Jain 1:22:43
You can also revert this back if in case you want to use pro.
OK, so now I'll show you two approaches. The first approach is we'll be using LCL. The second approach is we'll be using Landgraf. So when we define Landgraf, there are three things we need to define. One is state, which is your last. Then whatever functions you have, you will define that as a node.
So what functions do you think rag as? The first function that a rag as is retriever. So inside retriever what will you define? You have to keep it as state and then you just have to define DB equals to sorry context equals to.
DB dot your search technique whatever search technique is there and you will pass the user query.
And what will be your answer generation be? You will have function called generate. This will be state.
And then you will have answer.
Equals to DB dot not DB. It will be LLM dot invoke.
So this will have both query plus context. So looking at this two node. So what is this? Retriever is a function but when it comes to land graph state this is referred as node and generate is referred as node.
So what are the state variables that you're supposed to track? So when I define class, I need to have variables, right? So looking at this two functions, which are the three variables do you think we have to trace?
So in bank balance we had in bank this thing we had balance which was the attribute. Now here what attributes do you think we have?

Tirth 1:24:36
So we have to trace the documents.

Tarun Jain 1:24:39
So I'll define.
Class rag I will define. Inside rag I need to have 3 variables. Which are those?
Oh, can you repeat?
Uh, I didn't get you. Can you repeat?

Tirth 1:24:59
Box.

Tarun Jain 1:25:03
What dogs? No, not dogs.

Ajay Patel 1:25:04
Rocks, rocks.

Tarun Jain 1:25:09
Variables. So if you notice in answer dot PY we don't have docs.
So we are just using the client and once client is there we are only defining DB. There is no document that we are adding and I've already defined all the three variables here only the three variables are defined here.
So we have context. So what is context? The data type of context?
It will be list of STR.
And then we have query. What is query? It is a string. Then we have answer which is STR. Is this clear the variables that we want to persist in this state?

Ajay Patel 1:26:03
Mm.

Tarun Jain 1:26:10
So here when I define context, what I will what will I do? I'll just define it as state of query and when I'm defining this answer I will define state.

Of query and here it will be state of context.
Is this clear? Anyone has any doubt what I'm trying to do here? So first thing what we are trying to do is when we define land graph, you have to define a state and that state I don't want to directly come up with the variable, so I'm defining the logic. So the logic is nothing but your node. How do you want your rack pipeline to be? First I want to.
Retrieve the information and then I want to generate the information. So when I retrieve what are the variables I need? I just have a context and then if I want to extract the context from vector DB I need to have a query right? So in order to get the context I will just use DB dot maximum marginal relevancy and.
User query and here you will have KK equals to three. So now this context also needs to be persisted so that when I use generate I need to use it for query plus context. So how many variables I have now? I have state query then I have state.
Context and then this answer needs to be persisted. So total there are three variables, answer, context and query which I'm adding it here. Now this is my state.
Is this clear?

Ajay Patel 1:27:48
Yes.

TJ

Tarun Jain 1:27:51
So for rag, it is very simple. Rag will only have three variables, context, query and answer. Node is just function. Then what we have to do is we have to create edge. So edge is nothing but how do you want this nodes to be connected. So first will be retriever.
So I will start with retriever.
Then retriever.
Will pass to generate because generate needs context and context will come from retriever and once you have generate you will end it.
So edge is like OK, you have nodes now. How do you want the nodes to be connected from start to end? So we'll start with retriever which will get me the context. Then retriever will go to generate, then generate will go to end. So.
For first time if you see you might get confused but just make sure you follow this diagram. So if you see you get the retriever and then generate and once you have LLM you have the final response so you'll know what is the start and what is the end.

Is this clear? This is for Langraff.
And whatever I've written here is just a syntax.
The code is already there. If you see this code we have already written. This code also we have already written.
This syntax is clear. I'll proceed.

Hardip Patel 1:29:22
Yes.

Tarun Jain 1:29:24
OK, so first I'll use what you call. First I'll use NCL. So for LCL I need output parsers, so from Lan chain.
Core. We need output parsers, so I don't need Jason, so I'll just use the string output parser.
And then since I want to pass my input variables, I'll use from langchain dot core.
There is something called as runnables I need to use.
Runable pass through.
Where is it?
So why are we supposed to use this? When I'm giving the input variables, only context is defined. Context is defined in retriever. So if you see here there are two input variables, one is context and one more is query. The context is already defined and it is supposed to be coming from retriever. So in order to do this, what I will do is I will.
Runnable pass through. This is the only new function that we are writing today. Apart from that, whatever you saw earlier, it is the same things that we repeated last two days.
And how do I define output parsers? You can just define output parser or just parser equals to. Do you think I need to define a schema for a string output parser?
So are we supposed to define a schema?

Hardip Patel 1:31:03
It can be anything if it is training there.

Tarun Jain 1:31:07
For string it is not required, but if you need Jason we have to define a schema. So for

Jason what we usually define we define pydantic output parser.
I hope this is clear at this time.

Hardip Patel 1:31:25
Yes.

Tarun Jain 1:31:27
Cool. So now what I can do is I can just define the chain. So chain is the final step and what are we supposed to parse? One is the context.
Which is coming from retriever.
Sorry, it's not retrieval. What did we define?
So after text docs what you can do is you can define retriever.
Equals to test docs dot.
Sorry, it should be DB. Where is DB?
DB dot us.
Retriever and what is the search type? Search type is nothing but MMR and MMR stands for maximum marginal relevancy and search keyword arguments K equals to K which is 3. So you just have to note down this line under retrieval.

Hardip Patel 1:32:29
Done.

Tarun Jain 1:32:30
So where is it coming from? It's coming from DB. This is just for testing purpose.
For retriever, just define retriever equals to DB as retriever and what is the search I need? I need MMR. So if you need similarity you can just write similarity.
But I need MMR.
So you can write this above augmentation.
Is this done?

Hardip Patel 1:33:18
Yes.

Tarun Jain 1:33:19
OK.

So here context is nothing but retriever which will get me the information. Then you have in one more input variable which is query. This I will just use runnable pass through which a user needs to give. I don't have the input for this.
And then just pipe symbol. What is the first pipe?
It is prompt template.

Hardip Patel 1:33:44
Hmm.

Tarun Jain 1:33:46
And what is the second pipe? It is your LLM and 3rd pipe is.

Hardip Patel 1:33:48
Everything.
Hmm.
I said.

Tarun Jain 1:33:54
This logic everyone knows prompt template, LLM and parser. The only thing is these two variables it's coming from different thing, so I'm defining it in a runable pass through. So this logic is only for rag.
This one line you have to add, but usually you'll not use LCL, you'll be using Langraf.
And now once you have chained, what is the last function that we are supposed to run?

Hardip Patel 1:34:22
Invoke.

Tarun Jain 1:34:23
Just invoke response equals to chain dot invoke.
Query is.
Did I define query here?

Hardip Patel 1:34:37
You look great. Is it front?

Tarun Jain 1:34:39
Uh, I made it XYZ.
User query.
Everyone understood the flow till here it was the same thing till generation. After generation only for rag there are two approaches, one is LCL, one more is Langraf.
For Langraf the feature is you have to define a state so that you can compile the graph.
For that you have two nodes and for rank there are only two nodes. This is a basic template, one is retriever and one more is generate. Retriever will fetch the context, generate will generate the result and you have to define the workflow inside Edge.
So the three terminology are state.
Node and edge.


Hardip Patel 1:35:26
Mhm.

Tarun Jain 1:35:27
If this is confusing, we'll cover this tomorrow because Landgraf is very important.
LCL we have already covered. We just have to use five symbol. It starts with prompt, then LLM, then parser.
And since we don't have the query defined, but we have retriever defined, it's coming from context and then we can directly use an enable pass through.
And let me check how to add the callback. We have to add callbacks equals to Opic tracer when we do invoke chain.
If you just do print response, this should be string. We don't have to use response dot content.
Saying I don't know.


Hardip Patel 1:36:24
Mhm.

Tarun Jain 1:36:25
Contact details of.

Hardip Patel 1:36:28
I don't get anything.

Tarun Jain 1:36:40
Or this.
Run the product page.

Hardip Patel 1:36:46
Don't we? Don't we need to add the?
Retrieve chunks.

Tarun Jain 1:36:53
Let me check the parser if we are getting the context or not. So this is the chatbot. If I click the chatbot I have Google generative AI so if you see you are getting the context.

Hardip Patel 1:37:02
No.

Tarun Jain 1:37:07
So this is what Opic Tracer does. So Opic Tracer will track your.

Hardip Patel 1:37:09
Mm.

Tarun Jain 1:37:14
Duration, which is latency tokens and cost. Here if you see you have input and output. For input you are able to track the context. So for the given query, what was the question that I had? I'll click YAML.

Hardip Patel 1:37:24
Mhm.

Tarun Jain 1:37:32

So this is the content. You are the seasoned. This is system prompt.
And now if you look at the user prompt, you have the context.
And after context you have user query which is what is authentic product about.
Response it didn't.

Hardip Patel 1:37:53
Same.

Tarun Jain 1:37:54
OK, give me the response here.
OK, it's give the response. Do you have this product or is it wrong?

Hardip Patel 1:38:12
Yeah.

Ishan Chavda 1:38:12
Yeah.

Hardip Patel 1:38:16
Yeah, there's a there are these products.

Tarun Jain 1:38:19
So there is one thing that we do. We have to do here. Instead of K equals to three, just give K equals to two. I guess the Gemini again, it's hitting the limit. Instead of three, give 2.
But you can try with pro as well if you're getting the results or not.

Hardip Patel 1:38:42
Um.
I I'm getting results. I wrote who is on the team. I got the answer.

Tarun Jain 1:38:52
Oh, you're using the different data set, right?

Hardip Patel 1:38:54

No, I'm using URLs because then the docs didn't work, it went out of.
I'm using whatever URLs you are using. Uh, I'm getting good enough results, but Pro is getting better.

Tarun Jain 1:39:04
OK.
How many of you are using docs? Which one are you using? The document or website?
Anyone else is using document?

Hardip Patel 1:39:27
I was using document but it was not like home.

Ayush Makwana 1:39:29
Yeah, I am. I am using.

Tarun Jain 1:39:32
Have you got the results? Can you verify if it is right or wrong?

Ayush Makwana 1:39:35
So actually uh, I'm using a different PDF, so.

Tarun Jain 1:39:41
Ha, that's fine. You just have to test if you're getting the output which is correct from the PDF or not.

Ayush Makwana 1:39:46
Yeah.

Tarun Jain 1:39:49
And once you get the output, you have to trace it on opaque.
Where you have to see how much tokens you're utilizing and what is the cost of it and then also check the response here.
So if you keep in YAML you will be able to see system prompt. If you just prettify, you'll only see user prompt.

Ayush Makwana 1:40:04
Oh.

Tarun Jain 1:40:16
Plan that will cover tomorrow because two details today will be too much, but I hope you understood the flow.

Hardip Patel 1:40:16
Tarun.
Uh.
Yeah, I have one question regarding OPIC.

Tarun Jain 1:40:30
Oh yeah.

Hardip Patel 1:40:31
So like let's say we let's say we had to use multiple queries but it is just like one flow.
Can I combine multiple queries on on Opic to see the?
Aggregation or something like that?

Tarun Jain 1:40:51
Oh, combine in the sense.

Hardip Patel 1:40:53
I mean like a trace multiple multiple search query results under one ID. Can we do that or?

Tarun Jain 1:41:05
OK, you mean this three prompts that you have, it should come under one single key.

Hardip Patel 1:41:08
Yeah, yeah.

Tarun Jain 1:41:11

Oh.
OK, I don't think that will be doable because all these are unique, I mean unique entries.

Hardip Patel 1:41:17
OK.
Good. OK, fine.

Tarun Jain 1:41:26
So I don't think it will be possible.

Hardip Patel 1:41:29
OK, at least I can do it with tags tags, right?
I can tag it. So there are tags. Yeah, you can like we can add tag to it now. OK.

Tarun Jain 1:41:33
Uh, do it with.
ID.
OK, yeah, yeah, I it's OK, yeah, yeah. And this I'm not used it, so I'm not that much offshore.

Hardip Patel 1:41:46
OK, OK.

Tarun Jain 1:41:47
So the only thing that I've used is the feedback scores. Mainly we use it for debugs and then metadata if you're using tools. So like what did the tool return? Usually input and output will not track that, so for that if you're using any MCPS.

Hardip Patel 1:41:53
Hmm.
But.
OK.
Hmm.

Tarun Jain 1:42:05
During that time you use metadata. Apart from these three, I don't think I've used any other things.

Hardip Patel 1:42:10
OK.

Tarun Jain 1:42:11
These are not probably you can check documentation so but I'm not sure how this works. For me this is unclickable.

Hardip Patel 1:42:18
OK.

Tarun Jain 1:42:22
If I make that.
Rag.
Where did that go?

Hardip Patel 1:42:33
Be able to filter with tag tags, I guess that's one.
OK, we want to see.

Tarun Jain 1:42:42
OK, we'll have to see this.

Hardip Patel 1:42:45
That's fine. I I'll look at it.

Tarun Jain 1:42:47
Yeah, so everyone got the results. Just verify it if it is right or wrong.

Hardip Patel 1:42:52
I guess uh there is any problem you can check in teams.

Tarun Jain 1:43:01
I don't take renewal pass to query.
Is this query or is it question? Can you just see if you're defining a question or querier?

Hardip Patel 1:43:08
Yeah.

Tarun Jain 1:43:18
Uh, where is? Can you see if you have defined your query or question?

Tirth 1:43:25
He said he.

Tarun Jain 1:43:26
Query only.
Oh.
Why is it showing dictionary issue?
Oh, can you share your screen?

Hardip Patel 1:43:40
I think he might be using pedentic person.

Tarun Jain 1:43:47
No, it's string on. Can you one second? Can you Scroll down?
Query. Query. OK, can you scroll up?
No where you're defining query user query.
Scroll up, Scroll up.
OK, now how many levels retriever so?
Yeah, now can you Scroll down?

Tirth 1:44:24
Upon.

Tarun Jain 1:44:26
Don't don't don't.

Hardip Patel 1:44:35
Mm.

Tarun Jain 1:44:37
OK, can you Scroll down?
Shin dot.
One second.

Hardip Patel 1:44:58
So.
Um.

Tarun Jain 1:45:02
We use the query.

Hardip Patel 1:45:06
Mm.

Tarun Jain 1:45:08
Can you remove the space and check once?

Hardip Patel 1:45:20
Mhm.

Tarun Jain 1:45:24
OK, I hope you got the results right. Uh, Rama Krishna.
Not enough information provided. Can you check in your logs if you're getting the right context?
Or do you think whatever data you have used has that context?

Tirth 1:45:50
I'll put it expected so it works.

Hardip Patel 1:45:53
Or big checker or big.
If it has the relevant chunks as a context.

Tarun Jain 1:46:04
OK, so whatever we did as of now, it is what you call us plain traditional drag. So if you want to improvise this, the best approach is we use hybrid search. So what is hybrid search? We combine keyword search and vector search, right? Because most of the time what happens is.
Not just vector search will help. You should also have exact keywords and then you have to merge it. So that is 1 pending thing and then you have to rerank. So when do you think you are supposed to rerank?
Let's suppose your chunk size is only 1000 and you're defining K equals to seven. So now I know that every single chunk that I have has course, right? So what reranking will do is whatever 7 contexts that you have, it will rerank that automatically.
Based on the relevance. So for that we'll have models. So yesterday I showed Koyer, right? Koyer are the good players when it comes to re ranking. So we'll use re ranking in one of the approach and in one more approach we'll use hybrid search.
With payload because payload is very important.
But I hope you got the results. We'll try to improvise this on this using I grid search.

Hardip Patel 1:47:14
Good.
Alright.

Tarun Jain 1:47:23
So the code of yesterday's is available on the GitHub repo which is launch in quick start drag till here it was same till generation. After generation we added Opic then we added runnable pass through.
So R&A same only G was something new which you can check it out once which is

runnable pass through and the Opic razor. But Opic razor was also covered last week. Is this clear?

Hardip Patel 1:47:57
Yes.

Tarun Jain 1:47:58
OK.
So we need to have one land graph.
Example.
And.
Vector database.
Custom creation.
And how to handle the memory which is memory utilization?
For large vectors.
And then the advanced. Once we do this, the rack part is done, then we'll just have evals.
Valuation of the drug.
Cool. Uh, any questions?

Hardip Patel 1:48:59
No.

Tarun Jain 1:49:01
OK.

Ajay Patel 1:49:02
No.

Tarun Jain 1:49:03
Yeah, make sure you just have little bit practice on the rag because majorly it's the open source that we had the ugging phase one. That part the rag are some hands on experiences required, but whatever we discussed like Numpy pandas.
That's fine that EDA. If you do one or two times right, you'll get little bit of understanding, but RAG is very important RAG and agents.

At least in terms of the development that you guys are doing.
One.
Oh, that's it, right?

Tirth 1:49:40
Thank you very much.

TJ

Tarun Jain 1:49:42
Yeah.

Ajay Patel 1:49:45
Thank you.

Ronak Makwana 1:49:47
Yeah.
Good.

Ishan Chavda stopped transcription