Python and AI Power-Up Program Offline Class-20250903_114829-Meeting Recording

September 3, 2025, 6:18AM

1h 22m 6s

◉ **Ajay Patel** started transcription

Tarun Jain 0:04

Pass that topic to LLM as a output parser. What will be the format of Jason?

Tirth 0:06

Mhm, mhm.

Mhm.

It will have FAQ as an array.

Tarun Jain 0:13

OK.

FAQs as an array inside that what will we have?

Tirth 0:18

As an array of question and answer of question and answer.

I said.

As a string.

Tarun Jain 0:25

I will have question.

This is the value what LLM will generate and then I will have answer.

Tirth 0:36

Select.

Tarun Jain 0:41

So I said two or three, so this again I will copy.

Tirth 0:46

Mhm.

Tarun Jain 0:52

Then.

Tirth 0:54

And after FAQ there would be joke.

Tarun Jain 0:58

And then you love joke, which is in string.

Tirth 1:02

Yeah.

Tarun Jain 1:04

OK, so this is the pidantic class and what you have to do is you just have to guess whether you'll get the output or it is an error. So I'm using chat run template and I'll show both the example one is.

LCL with parser and one more LCL without parser. So without parser in the sense I won't use this parser here.

Tirth 1:26

OK.

Tarun Jain 1:27

OK, so this is the first prompt. You are an expert copywriter who has expertise in relevant jokes, writing and FAQs, writing, covering diversity, jokes and FAQs, writing, covering diversity.

Just one second.

Tirth 1:47

Mhm.

Tarun Jain 1:49

So now can you just give me one second?

Tirth 1:54

Welcome.

Tarun Jain 2:58

OK so here I just have a simple system prompt. Now I'm using the parser. So for parser what am I doing? I'm parsing the pydentic object which is fax topic.

Tirth 3:10

Mhm.

Tarun Jain 3:11

Which is this one.

And then the same thing in messages I'm passing system prompt and user prompt.

So what do you think will this particular response print?

Tirth 3:19

Mhm.

EA.

Tarun Jain 3:25

So you can just check from here chat prompt template to response. So do you think it will generate the output or will it be an error?

Tirth 3:37

Internet output, but I don't know like.

Uh, how many FAQ to in it? I'm not sure.

3:48

Um.

Tarun Jain 3:51

No. So what you can do is you can at least guess like will you get an output or will you get an error?

Tirth 3:58

You're giving topic AI users with topic and it should work.

Ajay Patel 3:58

Yeah, it is.

It it should work.

Tarun Jain 4:06

How many of you say it will work? All of you.

Tirth 4:06

Did you change anything you're doing between the parser?

Last topic.

I think it would work. We have messages and yeah, I I think it should work. Yeah, yeah.

Tarun Jain 4:18

OK, let me run this. It's an error.

Tirth 4:22

Bye.

Tarun Jain 4:24

So what do you think this LLM will return and what should be this parser?

Ajay Patel 4:29

LLM should return Jason, sorry this response and parser will parse the response according to the mod base model identic object which we have created.

Tirth 4:31

Hello.

Alicia.

Tarun Jain 4:40

But is the LLM have the schema?

Does the LLM what's might needs to send it? Because it's an error.

Tirth 4:45

It gave the response.

Ajay Patel 4:48

Yes.

Tarun Jain 4:51

So can anyone tell me what is missing?

So you should what I'm trying to say. LLM doesn't have the schema to generate into parser. That context is missing so.

Tirth 4:57

And.

OK, OK. So I think we gave to LLM yesterday like this is the schema you should follow.

Tarun Jain 5:13

So what is that keyword?

Tirth 5:20

Not yet, but does it give a passer don't generate?

Tarun Jain 5:21

Parser dot.

Tirth 5:27

Output get formatted instruction.

Tarun Jain 5:28

Get format instruction. So what should be here?
Format instructions.

Tirth 5:35
But.

Tarun Jain 5:38
So is this clear? Every time you define parser with LCL, what you are supposed to do is you need to have this key which is format instruction. Why? What is the purpose of system prompt role play?

Tirth 5:41
OK.

Tarun Jain 5:54
Instruction and output parsers. Output format in the sense it you are fetching it from parser dot get format instruction. Now LLM knows hey you need to generate the final response in this particular schema.
Right. And once that is done here what is LLM will generate. So you have prompt, you have LLM. Now LLM will generate this particular Jason and specific key.
So there might be chances LLM. If it doesn't have schema, it will generate it in this format, but it will make it as frequently asked questions or something even if it is in capital rate.

Tirth 6:38
Um.

Tarun Jain 6:42
FAQs.
Do you think this FAQ will match with the schema that parser have?
So parser is matching the schema that you provide, but if LLM itself doesn't know what is the schema, then how will it even parse that particular function?

Tirth 6:53
Mm.

Tarun Jain 7:00
Is this here?

Ajay Patel 7:01
Yeah.

Tirth 7:02
Basically.

Tarun Jain 7:03
So again, if I this, I'll get an error.

Tirth 7:04
But instead of Tarun, just asking yesterday we were not adding this to the system prompt.

Ajay Patel 7:12
Hmm.

Tarun Jain 7:13
No, we were.

Tirth 7:13
We did something else like we.

Hardip Patel 7:14
We were, we were adding it.

Tirth 7:17
OK.

Tarun Jain 7:18
I'll show the code, but before that I wanted to ask one more follow up question. I

added format instruction. Now even if I run all the cell I'll get an error. What should I edit?

Tirth 7:22
OK.

Tarun Jain 7:33
So I'll come to Thir's question, but before that you just have to answer this. I added format instruction now even if I run all these things.

Tirth 7:39
OK.
Mhm.

Tarun Jain 7:43
I'll still get an error.

Tirth 7:45
Oh, you have to give. You have to give the partial variables for setting this, yeah.

Tarun Jain 7:49
My name.
So here what I need to do, I need to add.

Tirth 7:52
I will see you.

Tarun Jain 7:57
Partial.
Variables.

Tirth 8:02
OK.

Tarun Jain 8:03

Format instruction. Let me check this spelling.

Mitesh Rathod 8:06

Mhm.

Tarun Jain 8:08

Parser dot get format instruction. Now this will work.

Tirth 8:11

Yeah.

Tarun Jain 8:13

So if I come back to yesterday's code.

Using parser with LCL pipeline in system prompt we had format instruction, then we define partial variables and then we had parser.

Tirth 8:25

No.

Tarun Jain 8:29

And here if you see we are just giving the review which is this particular input variable and the input variable that we have format instruction inside system from it is appended into partial variables which already has certain value.

Tirth 8:35

Mm.

OK.

Hey.

Tarun Jain 8:47

And now what is the response? So how do I get response into dictionary format? What is the function?

Tirth 8:58

Done petition.

Hardip Patel 8:59

Moderate them.

Tarun Jain 9:00

dot.

Mitesh Rathod 9:00

Moderator.

Tarun Jain 9:02

A what?

Hardip Patel 9:03

Not modeled answers.

Tirth 9:04

You're dumb, Jason.

Tarun Jain 9:04

Model.

Mitesh Rathod 9:04

OK.

OK.

Tirth 9:06

A model dump. Yeah, model dump.

Tarun Jain 9:10

So we have a fake use which is in list and then we have joke which is in.

String. Is this clear?

Tirth 9:17
Think format.

Tarun Jain 9:19
You understood the syntax. So LCL we need to have a key or you can say input variable which has format instruction and format instruction is usually appended in partial variables. Why? Because we have the value of it.

Tirth 9:20
Yes.
Yes.

Mitesh Rathod 9:25
We put.

Tirth 9:35
Yeah.

Tarun Jain 9:36
And then you can use parser. If you miss out parser here you will get Jason. Then what are you supposed to do?

Mitesh Rathod 9:46
Uh.

Tarun Jain 9:46
You need to use parser dot parse response.

Tirth 9:49
E.

Ajay Patel 9:49
Parts too soon.

Tarun Jain 9:51

Is this clear?

Tirth 9:53

Yes.

Ajay Patel 9:53

Yeah.

Mitesh Rathod 9:54

OK.

Tarun Jain 9:55

OK, so we have one more. We have LCL without parser. So this is the system prompt. Same we have sentiment analysis and then we have format instruction. Let me also copy.

Tirth 10:21

Yesterday we were not appending this. It was automatically appended. Either it was went from template or.

Hardip Patel 10:24

Pinpinvia Agent.

Mitesh Rathod 10:25

Play K.

Tarun Jain 10:27

Oh, which one?

Mitesh Rathod 10:28

So the egg well.

Hardip Patel 10:29

So in React agent we don't require this. We were not appending appending it in React agent, but for chat template from what was it in that we were using it.

Mitesh Rathod 10:32

Uh, we were not. OK, my phone checked. What was it in that we were?

TJ **Tarun Jain** 10:42

Now for structured response, we use the what you call input variables everywhere.
Format instruction then without LCL also we had parser variables and here if you see when you get the content right without parser we had response dot content which was in Jason format then we used parser dot parse.
We had partial variables everywhere. Only for React we didn't use partial variables.

Tirth 11:04

It.
OK.

TJ **Tarun Jain** 11:12

So second thing I'm using sentiment analysis. I have parser 2, I have system prompt.
System prompt is the same that we used yesterday. Here you have prompt template 2 which is taking system prompt 2 which is this one.
And now can you tell me what is the error in this?
If I run this, will it give me an error or will it work?

Tirth 11:37

Can you scroll up a bit at all? Can you give the partial variables for this one?

Hardip Patel 11:43

Yeah, I'm sad of you.

Tirth 11:47

We have to give the in validation system prompt, so we still have to give it as input variables. So this will go with review. We also give have to give the format instruction at the end.

Hardip Patel 11:49

I said.

Ajay Patel 11:53

Is.

Yeah, Format distance is missing.

Tarun Jain 12:00

Correct. Here what you need to do. Here you have to define.

Tirth 12:02

Format instructions.

Ajay Patel 12:04

Format instruction.

Tarun Jain 12:06

Parser 2 dot get format instruction. What is the better approach?

Tirth 12:11

We do it in the chat prompt template with messages.

Ajay Patel 12:11

Certainly.

Tarun Jain 12:15

Oh.

Is it clear? I hope there should be no confusion here because this concept is very important. Whether you use RAG or if you just use simple LLM call, this will be used everywhere.

Tirth 12:18

Yes.

Ajay Patel 12:19

Yeah.

Tarun Jain 12:37

Now, how do I get the final response in Jason?

But this auto complete is wrong.

Can anyone tell me the statement?

Hardip Patel 12:48

So.

Mitesh Rathod 12:50

Response to.

Ajay Patel 12:50

Uh.

Tirth 12:50

Can you? Yeah, we are passing it to LLM only.

Tarun Jain 12:52

Response to.

Ajay Patel 12:52

No, no response dot.

Tirth 12:58

The response got.

Tarun Jain 12:58

So think LLM rate. If I use LLM dot invoke and saved in response, how do I get this string?

Tirth 13:06

Response to dot content.

Tarun Jain 13:08

Now now what?

Tirth 13:10

OK.
So we can do parser dot parse uh with that content. It will remove dot Jason format and everything around it.

Tarun Jain 13:18

And then modeled them.

Mitesh Rathod 13:22

And.

Tarun Jain 13:24

Is this clear?

Ajay Patel 13:26

Yeah.

Mitesh Rathod 13:27

Yes.

Tarun Jain 13:30

What?
OK, this should be parser 2.

Mitesh Rathod 13:38

Um.

Tarun Jain 13:43
This is clear.

Tirth 13:43
Yeah.

Ajay Patel 13:45
Hmm.

Tarun Jain 13:46
I define 2 parsers. In this parser 2 I have the logic of sentiment analysis and this was sentiment analysis.
Is this clear?

Mitesh Rathod 13:55
Yes.

Tirth 13:55
Again.

Ajay Patel 13:56
Yeah.

Tarun Jain 13:58
OK, so let's get back.

Mitesh Rathod 13:59
Tarun yesterday I tried to check on the flash problem. The the flash was not using tool call the. I think I checked or whatever but it said that it is because it is in fast response mode.
Could it be the reason?

Tarun Jain 14:20

No flash here. It was working for me every time.

Yeah, if you see name it, pick web search.

Mitesh Rathod 14:28

I tried to use it two, three, three different type, like in ways, but I wasn't able to do it.

Tarun Jain 14:37

Using Flash.

Mitesh Rathod 14:39

Yeah, using Flash.

Because.

Tarun Jain 14:43

Uh, with pro you are able to get the results right?

Mitesh Rathod 14:45

Hmm.

Tarun Jain 14:47

With Pro you are getting the results.

Mitesh Rathod 14:49

Perfect results.

Tarun Jain 14:51

OK, then that's mainly the model issue.

So did anyone try this?

Mitesh Rathod 14:57

Yeah.

Tirth 14:57

Same with me. I was getting I I was getting good results with the pro, not with the flash. Same happened with me.

Mitesh Rathod 15:06

Yeah.

Tarun Jain 15:08

Did anyone try with this too? Tabli and Yahoo Finance?

Tirth 15:13

No, it was with another task.

Tarun Jain 15:16

OK, so I'll keep this here itself so we can have like one or two more days because still we are yet to do agents, right? So we still have some time to experiment with this. You understood it what we need to do here, so here in React.

Ajay Patel 15:16

OK.

Tirth 15:20

OK.

Mitesh Rathod 15:24

M.

Tirth 15:24

Yeah.

Mitesh Rathod 15:29

Yeah.

Tirth 15:30

Right.

Tarun Jain 15:31

You need to use two tools, one is web search and one more is Yahoo Finance. And you have to configure web search which is Tabley and Yahoo Finance here.

Ajay Patel 15:44
Mm.

Tarun Jain 15:44
Uh, where is Capdogo?

Tirth 15:45
It.

Tarun Jain 15:50
Similar to Doug Doug. So Langchain provides both Tably and as well as Yahoo Finance. Yahoo Finance doesn't need any environment variable, whereas Tably will need an environment variable for which we need to paste the API key, right? So if we have these things then probably what you call.

Tirth 15:52
M.

Tarun Jain 16:08
You'll have some understanding of how you can improve the routing and what is the importance of routing because there might be high chances. Let's suppose you use search and finance if you ask anything related to stock if it uses search. Technically it is correct, but in terms of implementation it is wrong because through search also you can get the result. But technically what tool is supposed to be picked? It's Yahoo Finance, right? So that routing is very important which you need to observe.

Tirth 16:33
OK.
Amex.

Tarun Jain 16:40

And that routing part will be covered when we pick agents.
So we still have this entire week, so we can experiment with this. So now what we can do is let's start with the vector database part and we'll be using fast embed similar to what we used last time. And then for vector database you'll be using quadrant.
I'll also list down some of the good vector databases.
Victor.
Database to pick one is you have quartered, you have deviate and then you have something called as milverse.
These are three good vector database that one can pick and most of these three things they have similar features. This is mainly on quantization. They're very good quantization and speed.
And Mirvus, it's good when it comes to very large documents.

Ajay Patel 17:35
Tarun your your screen is frozen.
OK, now I can see the screen was frozen, but now it is working fine quarter.

Tarun Jain 17:41
Uh.
OK, are you able to see this writing part?

Tirth 17:50
Yes, now we are able to see.

Ajay Patel 17:51
I'm very increased.

Mitesh Rathod 17:51
Yes.

Tarun Jain 17:51
Yes.
OK, so BB8 it's good if you want to self host it self host on private cloud.
It's very straightforward is what I felt and most of the features like quantization and all even all these three provides. But if you want to experiment with advanced RAG

right if you are purely building on RAG.
Purely using vector DB.
Only for RAG, then quadrant is a good option because they have their own in-house hybrid search approach. So you remember the BM25 algorithm that I mentioned.

Mitesh Rathod 18:36
Hmm.

Tarun Jain 18:37
So that algorithm is directly developed by quadrant so you can reuse that particular components. So you have BM 25 then BM 42. So this is where if you only want to use vector database for RAG then you can use quadrant but if you are experimenting with.
Different use cases like recommendation system, then face matching. Most of the people what they do is they have applications like you have a database where you have tons of face matching and you want to build an algorithm where a new person if it is encountered you want to match the face.
If it is existing in the database or not, here you don't need any LLM. So when I mentioned drag, you need an LLM. But for the use case that I said now face matching, you don't need drag, you don't need drag, you don't need LLM. So during that time vector database are very good in terms of searching.

Mitesh Rathod 19:27
Yeah.

19:27
Mm.

Tarun Jain 19:30
Right. So during that time, if you have very large documents, very large in the sense in millions, during that time you can use minverse.
Is this clear?

Ajay Patel 19:41

Here.

Yes.

Tarun Jain 19:43

All three are open source. Quadrant is open source, Viviate is open source, and also Milverse is open source and there are three ways to run it.

Ajay Patel 19:46

Mm.

Tarun Jain 19:54

Three ways.

To run the vector DB client.

The first approach is cloud, the 2nd is docker which is running local.

Ajay Patel 20:10

Mm.

Tarun Jain 20:11

And one more is in memory.

In memory is like let's suppose I'm using Collab. I will save my data inside a particular folder. If the Internet goes off, my data is gone and if you're using VS code and if you're using in memory, you will create a folder. That data is available only within that folder if you want to access it outside.

Folder you can't, right? So what happens in Docker? Let's suppose you're working on three different projects and you created three different vector databases. So all the things will be there in the Docker which is local and then you have cloud. So what we will do is we will experiment with both in memory and cloud.

Persistered.

Within the folder.

And this three applies to all three of them. You can use Cloud, Docker in memory for Quadrant, VV8 and also for.

Mitesh Rathod 21:05

Yes.

Tarun Jain 21:14

So can you install these two library fast embed and 19 quadrant and also the starting line?

Mitesh Rathod 21:15

Oh.

Tarun Jain 21:24

Just these two, the Lanchen and Lanchen community.

Ajay Patel 21:31

Finished frozen.

Again.

Tirth 21:33

No, it is visibility.

Ajay Patel 21:35

I'm not sure. Nothing is moving over here. Let me rejoin this meeting.

RamKrishna Bhatt 21:39

I think for me also it is coming into fractions.

Tarun Jain 21:44

Oh, what are you able to see now? Are you able to see these three lines with install?

Mitesh Rathod 21:44

Uh, I think, uh, it is.

Tirth 21:52

Yes, can you can you share this link in quick start dot?

Do you want us to write it down?

Tarun Jain 22:02

I will share it.
Is this clear? The vector database to pick and what are the different ways to run vector databases? One is cloud, then you have Docker and then you have in memory. We'll try to use all the three approaches so that whatever you feel is comfortable based on your use cases, you can pick those.

Ajay Patel 22:37
Hmm.

Tirth 22:37
Mhm.

Ajay Patel 22:39
Yes.

TJ **Tarun Jain** 22:45
And I'll tell which one to pick at which particular time.

Tirth 22:48
Understood.

Ajay Patel 22:49
OK.

TJ **Tarun Jain** 22:50
So let me know once this is.

Ajay Patel 22:52
One one question here like the apart from these three, I see lot of other vector embedded database also like Postgres and Postgres is also supporting. Then there is also one more Rust base.

TJ **Tarun Jain** 23:00
I used fine point.
Ha ha.

Ajay Patel 23:09

I forgot a name, but a lot of other databases also supporting vector database. So what are the I mean different from the existing database?

Tarun Jain 23:21

Right, so Pinecone is there. Pinecone is also popular as what quadrant VV 10 provides, but this is API based right? And it's not open sourced Pinecone. Now if we talk about Postgres, SQL, Mongo DB and then there is also Elasticsearch.

Ajay Patel 23:24

Mm-hmm.

Hmm.

Mm.

Tarun Jain 23:40

So they're not pure vector databases, right? So what features you you require, right? In terms of vector database, you can use simple schematic search, you can use simple what you call vector search, but there might be times you have to manage memory.

Tirth 23:45

Sure.

Mitesh Rathod 23:50

OK.

Tirth 23:56

Mhm.

Tarun Jain 23:59

Right, because now today what you can do is when you save your data, which is the PDF that you were using, you can just see how RAM will go. So from 1.3 GB it will touch, it will touch proper RAM and if you're using it on disk.

Tirth 23:59

Mm.

Tarun Jain 24:17

Even there you will see some difference. You can use or simple drag all these three details, but the only thing is the memory optimization part and whatever features are there. And there is also something called as sparse neural retrievers.

Ajay Patel 24:18

E.

Tirth 24:35

OK.

Tarun Jain 24:37

So this is similar to like you have Lan Ching, you have Lama index.

Ajay Patel 24:41

Mhm.

Tarun Jain 24:44

At the same time you also have Google ADK and you have open AI swarm.

Ajay Patel 24:50

Mhm.

Tarun Jain 24:51

So yeah, if you see these players are already into LLMS and they're building their agentic frameworks, but they're not consistent enough, right? What features are required? They won't build it properly, but whereas Lan Chin and Lam Index, they know what they're trying to build and they will have those features.

Similarly here, if you look at PostgreSQL and Mongo DB and even Elasticsearch, they're very good vector simple databases. But when it comes to vector database support in terms of flexibility in how to manage memory, how to add quantization, how to support different search techniques.

◉ **Ajay Patel** stopped transcription

◉ **Ajay Patel** started transcription

TJ **Tarun Jain** 25:52
So if you want to add your own flexibility, you can't do it here because they're not open source.

25:57
Mhm.

TJ **Tarun Jain** 25:59
And in most of the cases you'll have to experiment with retrievers. I took one example right yesterday. What is black hole? Black hole will definitely return some context even if it is not there in your data. So during that time you'll have to experiment with different retrieval techniques.

Ajay Patel 26:15
OK.

TJ **Tarun Jain** 26:16
So this support is available in pure vector databases compared to those databases which.

Ajay Patel 26:19
OK.
Additional database.

TJ **Tarun Jain** 26:24
But you want migrated into Vector DB.
But I've heard good feedback in post SQL as well.

Ajay Patel 26:35

Mhm.

Tarun Jain 26:36

I did install it once, but I saw thresholding and all, but I didn't actually use it in any POC, but we have used it once. Mongo DB I never used. Elasticsearch also I've never used.

Ajay Patel 26:43

OK.

Tirth 26:46

Make a apartment.

Tarun Jain 26:53

OK, is this installed?

Mitesh Rathod 26:56

Yes, yes.

Ajay Patel 26:58

Yep.

Tarun Jain 27:01

OK, so the first thing is we have to define our embeddings, so let's just import from line chain.

Community dot we have embeddings dot past embed import.

Past embed embeddings.

And I will copy it here. So which model did we use last time? Was it Gina?

Mitesh Rathod 27:32

Yes.

Tarun Jain 27:33

This on it.

So what is the dimension of Veena? Does anyone remember?

A what? It should be 768. Now once you run this it will download the models.

Mitesh Rathod 27:52

768.

Tarun Jain 28:00

You don't need any API key for this.

Do you guys use Mongo DB or Postgres SQL somewhere?

Mitesh Rathod 28:08

Yes.

Ajay Patel 28:08

Normal products, yes.

Tarun Jain 28:10

Which one?

Tirth 28:11

Yeah, we use Postgres a lot. Also Mongo DB a lot.

Tarun Jain 28:17

OK, then we'll try to have one rag. Advanced rag is the right using post SQL.

Tirth 28:24

It.

Tarun Jain 28:24

So that if in case your entire database is already in Postgres SQL, you don't have to migrate into coordinator or any other vector database.

Tirth 28:29

Right, right.
7.

Tarun Jain 28:39
And there probably I can show you this thresholding because thresholding is supported in PostgreSQL.
Threshold.

Tirth 28:47
Can you go back to the import statement what we're importing as of now?

Ajay Patel 28:50
Yeah, yeah.

Tarun Jain 28:51
Yeah, so from line same embeddings, sorry from line same community dot embeddings then dot fast embed fast embeddings.

Mitesh Rathod 28:52
Do you face it?

Ajay Patel 29:00
Import import first links. OK, I'm waiting.

Tarun Jain 29:03
So if you want to use any other provider, it's the same logic from line chain.
Community dot embeddings. The good ones are Gina. Let me check if Gina is there.
You have Gina and if you use any other players like Open AI, then probably you directly have to install Lan Chain Open AI.

Ajay Patel 29:21
Thank you.

Tarun Jain 29:28

Because they're migrated into Langchain Open YAI, but if you need to use it from community, either it is Fast Embed or JINA. If not, it is Sentence Transformers.

Tirth 29:30
Action.
OK.

Tarun Jain 29:42
Community.
And weddings.
This sentence, this one sentence transformers.
So this one will load model locally. For Xena you will need API key. For fast embed you need API key and if you want to use closed source ones you can directly use the library like Line chain open AI is there.
And if you open line chain open AI you have embeddings. So if you see you have embeddings on this open AI.

Mitesh Rathod 30:11
OK.

Tirth 30:17
Gina asked for the HF. Gina will ask for the HF token.

Mitesh Rathod 30:22
So.
Yes.

Tarun Jain 30:25
No, no. Jina will have their own Jina API key.

Mitesh Rathod 30:28
OK.

Tarun Jain 30:29
Because Gina is a provider, it's like a competitor.

Mitesh Rathod 30:35

Maybe.

Tirth 30:35

OK.

Tarun Jain 30:36

So they will need API key.

Tirth 30:39

OK.

Mitesh Rathod 30:40

Let's OK, you know.

Tarun Jain 30:42

Some of some models have made it open source. If they made it open source, if you want to use Zena models, you can use it from sentence transformers. So what is the difference right sentence transformers?

Tirth 30:48

Yeah.

Mitesh Rathod 30:50

OK.

Tarun Jain 30:56

As all the open source models.

Tirth 31:00

Check.

Tarun Jain 31:02

Of embeddings.

Fast embedders very limited. Now the reason is if anything is available in sentence transformers, you have to convert that into ONIX runtime to make it faster.

Tirth 31:16

Mm.

Tarun Jain 31:20

Either you can do it separately. If you have the knowledge of deep learning, you can convert the existing embedding model into ONIX runtime and then use it, which is completely fine. But if you don't have that particular knowledge, what we can do is we can directly use fast embed. So fast embed what it does is.

It selected some of the best models from sentence transformers and then it is converting it into ONIX runtime plus it provides data parallelism.

So what you see here which run in 17 seconds in sentence transformers, first it will install the CUDA dependencies and after CUDA dependencies it will install this model. So where is the model file? Here if you see you have Onix model dot Onix.

Tirth 32:01

Mhm.

You want to talk to.

Tarun Jain 32:10

But for sentence transformers you'll have model. So this dot after dot you'll have model which we saw earlier when we use tag in phase. So if you see here here it is 527 MB but in sentence transformers it might be in GB. That's the only difference.

Tirth 32:12

M.

Yes.

Mitesh Rathod 32:22

OK.

Tirth 32:22

Mhm.

Mitesh Rathod 32:24

Yes.

Ajay Patel 32:25

Mhm.

OK.

Tirth 32:30

OK, OK.

Tarun Jain 32:30

And Gina is a provider, so you will need API key. Same goes for Open AI.

Or Gemini. So if you want to use Open AI, the first step is very simple from.

Tirth 32:37

OK.

Tarun Jain 32:44

Blank chain open AI import open AI embeddings. I hope this spelling was correct.

Mitesh Rathod 32:55

Mm.

Tarun Jain 32:55

O is capital, A is capital, I is capital, A is capital.

So we just have to import this and then just instantiate embeddings equals to open AI embeddings and the model name will be text large.

Oh.

Mitesh Rathod 33:14

Don't have a baby in stock.

Tarun Jain 33:16

Oh, what?

Mitesh Rathod 33:18

Open blank.

Tarun Jain 33:20

Have you have to install here? I'm just showing syntax.

Mitesh Rathod 33:22

Oh.

Tarun Jain 33:27

So if I check their embedding models.

This is the embedding model text embedding 3 large.

You just have to copy that here.

Tirth 33:44

OK.

Tarun Jain 33:44

By default it will be the smaller one, but we have to use this thing. So these are the two models. One is text embedding 3 small and one more is text embedding 3 large.

Is this clear?

Ajay Patel 33:57

Yeah.

Tirth 33:57

Yeah.

Tarun Jain 34:11

And you can use the same function embeddings dot.

Embed query Hello world.

You'll have the vectors. So what should be the length of this?

Length of I'm getting that where it should be 768. Is this clear?

Tirth 34:28

It should build in.

Tarun Jain 34:37

OK, so this is it from the embedding part. So the only thing what you need to understand is if you need any providers you can directly use their import statements. And if you want to look at the open source one, then first preference will be fast embed if it is supported in this particular models, but if you need better models. Then directly you can use sentence transformers. This is only applicable if you have any use cases where you're restricting yourself to use only open source. If you're looking for performance, then definitely Open AI LLM and Open AI embeddings.
Now how do you pick?
Which open source LLM to use? Open source embeddings?
Uh, does anyone remember the leaderboard name?

Mitesh Rathod 35:23

Yeah.
Leader bird name.

Tarun Jain 35:25

For a meetings, there was a leaderboard.

Mitesh Rathod 35:29

Mhm.

Tarun Jain 35:32

There is something called as MTV leaderboard.

Ajay Patel 35:35

Mm.

Mitesh Rathod 35:41

Yeah.

Tarun Jain 35:43

I will also attach it here.

Mitesh Rathod 35:53

Again, it's.

Tarun Jain 36:03

Leaderboard.

For embeddings.

So here if you see most of the embeddings models will be there. What you need to do is you have to scroll on the right side. You will see this course retrieval, right? Retrieval is important. Relanking is important. Other things is not relevant for RAG.

Mitesh Rathod 36:31

M.

Tarun Jain 36:37

And one more thing, what do you have to understand is?

If your use case has English and as well as other languages, you will have to check if this particular embedding model supports that language or not. Whereas Open AI, Gemini, all these providers will support multilingual but open source that will be limited and most of the leaderboard scores that you see here.

Tirth 36:47

Mhm.

Tarun Jain 37:02

In this particular leader board you will see Chinese models. One is a Chinese model. GTE is a Chinese model. If you see GTE Quento, wherever you see GTE right, it is by Alibaba and 20% of the embedding models on this leader board belongs to Alibaba cloud.

Mitesh Rathod 37:10

Yeah.

Ajay Patel 37:15

Mhm.

Tarun Jain 37:21

And they have garbage results of Chinese tokens.

So that is something that you'll have to be aware of.

Tirth 37:29

So for multi language, for multi language, how do we get it? How do we know?

Tarun Jain 37:34

First thing is you'll have to check their model name. So if you see you have something called as multilingual E5 large instruct. So this supports multiple languages, but again it won't be suitable for Indic languages.

Mitesh Rathod 37:42

Blame.

Tirth 37:49

OK.

Tarun Jain 37:52

I don't know if uh has released anything or not embedding model.

Mitesh Rathod 37:52

Mr.

Ajay Patel 37:59

It was what I'm checking servo M.

Tarun Jain 38:02

OK, Sarvama has not released any embedding models.

They only have models.

Tirth 38:13

OK.

Tarun Jain 38:16

So if in case you have any use cases which has Indic or any Indic languages, then the only option is Gemini and Open AI.

Tirth 38:25

Jimmy M Bidding 001.

Tarun Jain 38:28

Huh, Jamie now is good.

It's at the top.

Tirth 38:35

M.

Tarun Jain 38:36

So if you look at the retrieval, so definitely when it comes to leader boards, right, you will have Chinese model to be tough, but the results sometimes are not that great.

Tirth 38:43

Maybe.

Mitesh Rathod 38:52

OK.

Tarun Jain 38:52

If you see you also have multilingual here, can you check this part? You can just click on multilingual.

Mitesh Rathod 38:55

Yes.

Tarun Jain 39:08

And now it only shows for English.

Now if you click on multilingual.

Gemini is multilingual, Quentry is multilingual, but mainly Quentry will be English and Chinese. GTE will also be English and Chinese GTE. Again, it's from Alibaba. This is also GTE.

Mitesh Rathod 39:32

There is a language specific cause in in there there is in in big option.

Tarun Jain 39:39

Where?

Mitesh Rathod 39:40

And the left side the.

And the second under the multilingual language specific option.

Tarun Jain 39:50

OK, this thing.

Coyar is there. Coyar, I forgot. Yeah, Coyar are also good players. So Coyar, if you want to directly use it, they have their model called AIA. So AIA supports around 128 languages.

This one. Can you see this higher?

This supports like more than 100 language something.

Ajay Patel 40:23

Mhm.

Tirth 40:24

OK.

Tarun Jain 40:24

They did mention 130 somewhere. Oh, 101 languages, yeah.

Ajay Patel 40:30

Thank you.

Tarun Jain 40:32

So Coier has both model and as well as embeddings. So the players in embeddings are Coier then.

Tirth 40:33

Even.

Tarun Jain 40:45

Oyer is not Chinese.
Oh yeah, then Gina excluding. I'm just writing excluding.
Open AI and Google because we know that Open AI has their embedding model and Google also have Gemini. But if you are want to explore different then you have Koya Gina then there is no Mic.
And here also we'll see Nomic somewhere Nomic AI.
No make.
Here also you have nomic. All this BGEGTE belongs to Chinese. So if you see here you have Chinese.

Tirth 41:30

OK.

Tarun Jain 41:35

What else is there?

Tirth 41:39

Inf Retriever.

Tarun Jain 41:39

Let me see.
Snowflake is worst.
I don't know how they have good benchmark.

Tirth 41:51

If we sort by retrieval, then we get Voyage 3.

Recently, yeah.

42:04

Mhm.

Ajay Patel 42:07

Link Voice 3.

Tarun Jain 42:15

Embedding providers.

Ajay Patel 42:17

Please.

Tarun Jain 42:23

Open AI Gemina is there. Coyer is there. Nomika as I mentioned.

Tirth 42:29

So here is there.

Tarun Jain 42:32

Voyage. OK, I've not tested voyage.

I'll see. I've not tested voice, but I've tested Coier. Coier has IR. We use this data set somewhere, so I'm familiar with it. But still I'll place Gina at top because they're the actual players when it comes to embeddings.

Tirth 42:43

It.

OK.

Mhm.

Tarun Jain 42:57

Nomic is fine for your user voice I need to test.

Tirth 43:10

OK.

Mitesh Rathod 43:12

Sweet.

Tarun Jain 43:12

And I didn't mention Azure Open AI because in Azure Open AI they don't have their own models, they're just using the Open AI models. So in LLM also I never mentioned Azure Open AI. If in case you have this founders thing right, I guess founders.

Ajay Patel 43:17

Mhm.

Tirth 43:19

OK.

Ajay Patel 43:20

Mhm.

Tarun Jain 43:29

Microsoft.

Have you registered for this?

Tirth 43:40

No.

Tarun Jain 43:40

So you'll get around $150.00 of credits. So using this you can use open AI's any model or any embedding. You'll also get what you call Azure portal for free.

Mitesh Rathod 43:43

Yeah.

Tirth 43:53

What's 15?

Mitesh Rathod 43:55

No, no. We get $1000 first, then 5000, then 15,000, then.

Ajay Patel 43:55

Yeah.

Tarun Jain 43:55

150K, sorry.

You should apply. This is very useful.

Tirth 44:07

So I was spending on open AI models directly. I don't have to do that.

Tarun Jain 44:12

So is it free or?

Tirth 44:14

No, no, I was. I I spent $2.5 yesterday.

Mitesh Rathod 44:15

How's it?

Tarun Jain 44:20

No, here you will get it for free. You'll also get Azure for free.

Tirth 44:23

OK.

OK.

Tarun Jain 44:25

Founders Microsoft.

And it's not like you'll get rejected. It is easy selection. I just applied for some random project I got selected.

Tirth 44:35

Mhm.

Mitesh Rathod 44:39

You just have your domain.

Tarun Jain 44:39

Just once we can show it.

Mitesh Rathod 44:46

Are you?

Yeah.

Hello.

Tarun Jain 45:00

Hello.

Mitesh Rathod 45:01

Personally.

Tarun Jain 45:02

OK, just one second.

Mitesh Rathod 45:09

Domains are registered. They don't ask for anything else.

Tarun Jain 45:30

Yeah, I'll share my screen again.

Word.

Are you able to see this?
So we started off from here, then it keeps on going going. They don't give you $150.00 directly. So after you spend certain amount they'll give you. If you see we spent this much of amount for free.

Mitesh Rathod 46:02
Yeah.

Tirth 46:10
Mhm.

Ajay Patel 46:17
Mhm.

Tarun Jain 46:18
So you should apply for this. You'll easily get selected. So from this what you can do is you can use Azure LLM and embedding.
So they don't have their own LLMS. So whatever you have in Open AI right, which is GPT 40 and GPT 4.1, you can deploy it on Azure and you can use it credits.
And it's very straightforward to apply.
OK, let's get back. Isn't done the embeddings part.

Mitesh Rathod 46:53
Yes.

Tirth 46:54
Yes.

Tarun Jain 46:55
OK, so now what we can do is uh, first thing is we have to import.
From.
Banking.
Quadrant import.
You have quadrant vector store and now what we also need is we need quadrant client import.

Blank.

Sorry, it should be wasn't client, huh?

So I'll show 2 approach. We'll use all the three approaches. One is cloud, one is Docker and one more is in memory. Today we will look at in memory. When we work with Rack, which is tomorrow, we will work with cloud.

Mitesh Rathod 47:53

Mm.

Tarun Jain 47:55

And only for one project I'll use post SQL just because you have your data saved in post SQL, but the flow is same if you're using line chain.

So instead of quadrant you will just have Postgres SQL and you will import Postgres vector data store and here it will be different. So what we are trying to do.

You want to save want to save the data inside Vector DB. First you need to create a project which is a collection.

Mitesh Rathod 48:21

OK.

Tarun Jain 48:33

So what we are trying to do is let's suppose I define collection name.

As if you remember, I used web pages right? So I'll just add chat chat bot. So in the initial version of this collection name I only have 4 web pages.

Mitesh Rathod 48:45

Mhm.

Tirth 48:50

Mhm.

Tarun Jain 48:55

Correct. Now once I save my data inside this particular collection name, you don't have to save your data set again, you can just read it. So this concept is called indexing.

Ajay Patel 48:56

Mm.

Tirth 48:56

Mhm.

Mhm.

Tarun Jain 49:09

Indexing should happen only once.

And once it is saved, it's done. And if you want to update it, you can update by adding new documents. And now let's suppose usually you will make it as data version one and you will index it. Hey, this is my initial version of the data, just save it. Save it and now if you want to update your chatbot with the new version of the data, you will create a new collection name saying data V2. But if you think you want to use the same data set then you can append.

Tirth 49:34

Mhm.

8.

Tarun Jain 49:42

So this is your call like whether you need to create a new collection or you want to reuse the same collection name.

Tirth 49:43

OK.

Tarun Jain 49:50

So collection name should be unique.

Should be unique. So now what is happening? I want to send my data inside Vector DB. I want to create this collection. Where do you create this collection inside a client? So client will have all the credentials. If you are using Postgres SQL you will have your DB name, you will have DB.

Password and you will also have one additional parameter like host right? So those

are credentials. Same goes for quadrant. If you're using cloud, not quadrant, but even Vivet, you will have quadrant TPI key.
And you will have endpoint URL.
So what is this endpoint URL? Basically whatever vector database you're saving it in a cloud, they will have their endpoint either saved it in AWS region, AWS at some specific region which is US or it can be GCP or it can be Azure.
So once you create any free cluster on cloud, you will get an endpoint URL where it is hosted with the port. So you just have to use those two variables and there we will create the client. Is this clear? So first we create a client with the credentials.

Tirth 51:07
Yeah.

Tarun Jain 51:11
Then we save that in a vector DB. So when we create this vector DB we have to define client if you see the first variable.
And the second variable is collection name.

Ajay Patel 51:21
Good.

Tarun Jain 51:24
So client is nothing but hey, this is the collection in that I've created which is a dummy space. Now since it is configured, whatever data you want to save, you can save it here.
Oh, is this clear the concept why we need client and vector store?

Mitesh Rathod 51:39
Yes.

Ajay Patel 51:39
Yeah.

Tarun Jain 51:40
OK, so first thing is I'll define a client.

Quadrant client and as I mentioned there are three approaches. If you see location is there URL. URL is mainly used for cloud by default quadrant users 6333 as their end point and if you Scroll down you will also find API key.

Right, so these are the three points. One is URL, one more is port and one more is API key for cloud. If you are using Docker then you will have location. Location is nothing but your local host 6003 three three.

Tirth 52:05

OK.

Tarun Jain 52:19

Here what I will do is I will just define a path.

Tirth 52:20

M.

Tarun Jain 52:23

And path will be a temporary folder and it can be any name. I'll just keep it.

I'll keep it DB.

So now if I do a list.

A list of temp DB. It should be empty, not empty. There is no file directory currently involved and now if I run this particular line and now if I run this.

Mitesh Rathod 52:56

Just.

Tarun Jain 52:56

It created meta dot Jason for it. If I run this again, it will throw an error. Why?

Because this already exists now.

Tirth 53:04

Um.

Tarun Jain 53:05

So which approach we are currently using? We are using in-memory.

Tirth 53:10

It.

Tarun Jain 53:11

If you are using VS code, what it will do is within the project directory that you are working only the data will be saved.

If you're using VS Code and let's suppose you define a the chatbot, it will create this particular folder and it will stay only there which is in memory. Once you delete that folder, your data is also gone. But in docar and cloud it is a bit different. You can save all your.

Client or collection name in the given local host or in the cloud. In memory it is persistent. Once you delete your folder it's gone.

Tirth 53:48

It.

Tarun Jain 53:50

Till here, is it clear? OK, so now what we can do is we'll use from line chain.

Ajay Patel 53:53

Hmm.

Tirth 53:53

It is clear.

Tarun Jain 54:00

I need documents community dot web base loader.

Import the base loader.

Mitesh Rathod 54:15

OK.

Tarun Jain 54:23

So what is the syntax after the input?

We have to define the loader.

And loader should have a web path.

Which is nothing but.

Mitesh Rathod 54:43

You are.

Tarun Jain 54:45

Did I write this correctly?

Huh.

So just notice here you just have meta dot Jason as of now. Once you save your data, you will have a new folder called collection and what will be that collection? Athenic chatbot.

Let me know till here if you have written it.

Mitesh Rathod 55:09

Yeah.

Tirth 55:09

And here it is done. We're working on 14th.

Tarun Jain 55:12

Data equal to loader dot load.

We just have four more lines of code, then probably we'll wind up today bitterly.

OK, why is it taking time? OK, it generated. So if I do the length of data, it should be just one.

And then data of zero I have total 2 fields. One is page content.

Tirth 55:45

Um.

Tarun Jain 55:51

And then one more is metadata.

So the next step is from Lanchen.

Expletors import.

Recursive character text splitter.

Splitter equals to.

So you remember the syntax?

Mitesh Rathod 56:19

Yes.

Tarun Jain 56:19

We just have to import the character text splitters. Then you have chunk size, then chunk overlap. I'm keeping it 0 because I don't need any repetition between the chunks and then splitter dot split documents. Why split documents? Because the data type of data is documents.

So I'm using split documents. If it is a simple string I will use split text, but in our case it is document. So split document and just add data.

So now if I do length of chunks.

It is done.

Mitesh Rathod 56:55

Oh, I did.

Tarun Jain 56:58

Just let me know if the is done. I'll delete this line. This was just to test. Just confirm you only have meta dot Jason.

Tirth 56:59

M.

Yeah.

Tarun Jain 57:09

It will create.

Ometa dot Jason file.

Initially it was empty.

So first one is client. This has to be done only once. If the folder already exists it will throw an error and then we are just loading certain documents so that we can save it in a vector DB.

Is it done?

Mitesh Rathod 57:52

Yes.

Ajay Patel 57:52

No, just a second.

Tirth 57:53

Yes.

Tarun Jain 57:55

OK.

So does anyone recall what is the distance measurement we have to use now for similarity? What is the distance algorithm?

Uh, what sound similar?

Tirth 58:15

There were two cosine. Yeah, cosine similarity.

Tarun Jain 58:20

I will delete this cell. This was again just to test.

Mitesh Rathod 58:23

Yes.

Tirth 58:24

M.

Mitesh Rathod 58:26

Yeah, done.

Tarun Jain 58:28

Is it done?

Tirth 58:30

Yeah.

Mitesh Rathod 58:31

Yeah.

Tarun Jain 58:31

OK, So what we need to do now is we created a client. We have to tell a client that hey we have a collection name, so can you create a new collection so from quadrant?

Tirth 58:32

Yep.
Mhm.
Uh.

Tarun Jain 58:48

Blend dot.
It should be HTTP dot.
Models.
You have distance.
And if we are using hybrid search, we need both vector params.
And we need spots.
So I hope everyone knows the difference between vector params and sparse vector params. So this is dense vectors, this is sparse vectors and what does sparse vectors contain?
We will not do sparse vectors today, but I just want to revise few concept. So what is sparse vectors?
But this.

Mitesh Rathod 59:42

Oh God.

Tarun Jain 59:45

Are you seriously?

Tirth 59:45

Dense vectors can go beyond one.

Tarun Jain 59:49

It is 0 or non 0 values.

Tirth 59:51

In Sparks victim.

Mitesh Rathod 59:52

That is always enough.

Tirth 59:54

It's nonzero values, right? Dense vectors can.

Mitesh Rathod 59:56

OK.

Tarun Jain 59:56

Algorithm. At least you remember algorithms.

We built recommendation system giving a hint.

Tirth 1:00:08

So we created that matrix of each with each.

Tarun Jain 1:00:11

What is that metrics called?

Tirth 1:00:14

I didn't take it. No, no.

Mitesh Rathod 1:00:14

Last night, please. Thank you.

Tirth 1:00:19

It wasn't very good on the chicken.

Tarun Jain 1:00:20

So let's suppose we have documents. We have 4 sentences. If that particular word exists, we added 1010, correct? That is our sectors. Now once you calculate everything, what did we do?

Tirth 1:00:24

Right.

Select.

Mitesh Rathod 1:00:30

Ask me please.

Tirth 1:00:32

Mhm.

Tarun Jain 1:00:35

We calculated the.

We calculated their dot product and then magnitude.

Tirth 1:00:42

Hmm.

Right, right. Not the magnitude a dot B. Yes.

Tarun Jain 1:00:44

Right. This is for distance measurement.

Now that comes for dense vectors, which is your cosine.

For algorithms, if you remember, we use an algorithm called TFIDF. First we calculate what is the frequency of occurrence of that particular word, and then we check how many times it was occurred in that particular document where we apply a lock function.

Mitesh Rathod 1:00:58

Mm.

Tirth 1:00:59

Ha.

Tarun Jain 1:01:12

You guys recall TF IDF and then you have PM 25.

Tirth 1:01:14

Yes, yes, yes, yes.

Mitesh Rathod 1:01:15

Yeah.

MBM 42.

Tarun Jain 1:01:17

So whenever you define sparse vectors, you need to have vectors which will return 0 or nonzero values, right? And the algorithms are TFIDF and BM 25 dense vectors is cosine so.

Tirth 1:01:18

And continue.

Um.

Tarun Jain 1:01:33

If you don't recall how this works, that's completely fine now. The only thing what you need to remember is if we are using dense, I will use cosine. If I are using sparse, I will use TFIDF for beyond 25. The only things you need to remember now is these two lines.

Tirth 1:01:46

Mhm.

Mitesh Rathod 1:01:49
Mhm.

Tarun Jain 1:01:50
So if you use dense vectors, you have to use vector params. If you use if you need both of them, you will use vector params and sparse vector params.

Tirth 1:01:50
OK.

Mitesh Rathod 1:01:52
Yes.

Tirth 1:02:00
Um.

Tarun Jain 1:02:01
You can just take a note of these two things because from now on this will be very important.
OK, so I've just commented this out. We'll look at these sparse vectors when we look at hybrid search, which is advanced rag.
This line is done from portent end dot HTTP dot models import distance and vector params.

Ajay Patel 1:02:21
Hmm.

Tirth 1:02:22
Yeah.

Tarun Jain 1:02:27
And now we just have to create client dot create.
Collection.
And you have to define collection name equals to collection name and whatever it

has generated that is correct. You have vectors configuration. What is the vectors configuration here? I have vector params.
And what is the dimension size?

Ajay Patel 1:02:55
2048.

Tarun Jain 1:02:55
768, which is already defined here.

Mitesh Rathod 1:02:57
16.

Ajay Patel 1:02:57
This one.

Tarun Jain 1:03:01
The length of embedding embed query hello world which is 768. You just have to copy that, paste it here. And what is the distance measurement? Distance equals to distance dot cosine. You will also have Euclidean.

Mitesh Rathod 1:03:15
Thank you.

Tarun Jain 1:03:17
But I don't need Euclidean. What else do we have? Yeah.

Mitesh Rathod 1:03:20
So like we are creating this embedding, so that's why we are using dense vectors. If we are searching then we will be using sparse vectors, right?

Tarun Jain 1:03:31
Yeah.
So if you are just using keyword based search, what you will do is here you will have.

Mitesh Rathod 1:03:38

Yeah.

Tarun Jain 1:03:42

Sparse conflict.

Then you'll have sparse vector params. So for sparse vector params you don't have to give anything like this because there is no fixed size for sparse vectors and but what will be your vectors? It will be VM 25.

Mitesh Rathod 1:03:55

Vent.

Yeah.

Tarun Jain 1:04:01

So whatever vectors you have, that vector configuration you have to provide. For vectors configuration it is from fast embed. For sparse it is either TFIDF or beyond 25. You just have to define the algorithm, that's it.

Mitesh Rathod 1:04:13

Next.

OK, just.

Tarun Jain 1:04:20

So you just have to remember these two lines. For dense vectors you have cosine, you also have Euclidian, but it's not recommended. This is it won't even make sense. Just keep cosine.

To avoid confusion for sparse these three things.

And the most famous one is BM25.

Now if you run this, the next thing is you have to do LS temp dot app.

OK, DB.

Mitesh Rathod 1:05:05

Yes.

Tarun Jain 1:05:06
Now if you see you have meta dot Jason then you have collection. Now if I print collection.
You should have authentic chatbot. As of now, this authentic chatbot is empty.
Whatever you have inside this is empty. Why? Because I didn't save my data yet. You understood the flow.

Mitesh Rathod 1:05:24
Yes.

Ajay Patel 1:05:25
Yes.
Yes.

Tarun Jain 1:05:25
I want to see my data inside collection, so I'm creating a space for it. So here you're just creating the empty space.
Let me know till here if it is done.

Mitesh Rathod 1:05:40
Yes, yeah.

Tirth 1:05:40
It is a storage dot SQL light. It is a storage dot SQL light. Is that right?

Ajay Patel 1:05:40
Yes.

Tarun Jain 1:05:41
Make music true.

Ajay Patel 1:05:46
Yeah.

Tarun Jain 1:05:46

Yeah, yeah, yeah.

Tirth 1:05:48

OK.

Tarun Jain 1:05:48

So most of them use a SQL at itself. There is also something called as Chroma.

Ajay Patel 1:05:52

In memory.

Tirth 1:05:55

Roman TV.

Tarun Jain 1:05:56

But it's very slow and no one uses it for production.

Tirth 1:06:01

I see.

Tarun Jain 1:06:02

Here also they use SQL Lite.

Tirth 1:06:06

Mm.

Tarun Jain 1:06:08

OK, why did they mention faster screen?

This is slow. If you use this, you'll get to know.

Mitesh Rathod 1:06:18

The fastest way to build.

Tirth 1:06:18

It is fastest way to will not execute.

Tarun Jain 1:06:20

Obviously in terms of range of code file. If you see here also it's the same logic. Use any vector database. The logic is straight. You have to start with client. Then first you have to create collection. Once you create collection you just have to add the data and then query it.

Ajay Patel 1:06:23

But.

Mitesh Rathod 1:06:24

It.

Tirth 1:06:27

M.

Tarun Jain 1:06:36

Obviously this is fast, but this line is very slow.
Compared to other LL compared to other vector database only this line.

Mitesh Rathod 1:06:44

Yes.

Tarun Jain 1:06:48

But even this is very straightforward, I don't think.
This is one line of code, then two. Now we just have to add two more lines of code.
Till here is it done. So first define the client, then create collection and now we need to create the vector store.

Tirth 1:07:01

Yeah, yeah.

Mitesh Rathod 1:07:02

Yes.

Ajay Patel 1:07:02

Yeah.

Tarun Jain 1:07:14

Vector store equals to quadrant vector store.

Mitesh Rathod 1:07:17

Client.

Yeah, absolutely.

Tarun Jain 1:07:21

And here you have to define client equals to client. Then I guess it should have embedding some collection, collect the name.

Mitesh Rathod 1:07:29

Collection name.

Tarun Jain 1:07:32

Equals to collection name.

Mitesh Rathod 1:07:35

And M.

Tarun Jain 1:07:36

And then embeddings. That's it.

Oh wait, it should be. It should be embedding. This is embedding.

Mitesh Rathod 1:07:45

Yes.

Tarun Jain 1:07:51

And now you just have to do vector store dot.

Add documents your chunks.

So here also they used add.

Tirth 1:08:07

I would.

Tarun Jain 1:08:10

What happened?

Tirth 1:08:13

Embeddings is not working. Let me check what I did wrong.

Mitesh Rathod 1:08:19

So we have meetings.

Tirth 1:08:19

Existing quadrant collection is done.

Tarun Jain 1:08:20

Did you get this token should be total 1012345678910. These are chunk IDs for all your chunks.

Mitesh Rathod 1:08:34

Yes.

It's still running.

Tarun Jain 1:08:38

I should take roughly. You guys are using the same thing, right?

Mitesh Rathod 1:08:42

Yes, yes, yes, yes, following the same.

OK, done. Done.

Tarun Jain 1:08:46

It should take less than 30 minutes.

Tirth 1:08:47

What did I got? What did I got? Like a 384 dimensional selected embeddings are 384 dimensional. Did I select the wrong one?

Mitesh Rathod 1:08:51

Have to invite.

Tarun Jain 1:08:52

No, no, no. Use 768. So let's suppose.

Did you define any name here?

Tirth 1:09:02

Yes.

Gina embeddings V2 base EN no model name. Oh model. I gave model. OK, OK, OK.

Tarun Jain 1:09:04

You give modeling or modeling?

OK, so now what has happened? You have picked this thing. By default it picked VGE small which has 384.

Mitesh Rathod 1:09:16

Yeah.

Tirth 1:09:16

Uh, I see.

Tarun Jain 1:09:18

So if you have to make that work here, you have to give creative work.

Tirth 1:09:23

I see.

Mitesh Rathod 1:09:23

Mhm.

Tarun Jain 1:09:24

384 is fast, but the retrieval is not that great.

Mitesh Rathod 1:09:25

No.

Tirth 1:09:29

Hmm, understood.

Ajay Patel 1:09:30

Mhm.

Tarun Jain 1:09:30

So 768 and 1024 is a standard value.

Tirth 1:09:34

OK, OK.

Mitesh Rathod 1:09:34

You will have to change the collection name.

Otherwise.

Otherwise.

Tirth 1:09:38

Got it.

Tarun Jain 1:09:45

So till here did it work?

Mitesh Rathod 1:09:48

Yeah, yes.

Tarun Jain 1:09:49

OK, so now what are the next step? We just have to test if it is working or not.

So what can I ask here?

Ajay Patel 1:09:58

Ready.

Tarun Jain 1:10:00

I'll just ask how do I contact Atyantik?

How do I find that?

Tirth 1:10:13

OK.

4.

Let me have.

Tarun Jain 1:10:19

So you just have to use results equals to.

Vector store dot This is what I was telling like if you look at other elements right you will not have this much of option. Sorry not elements. I meant vector database. You have maximum marginal relevancy search.

Tirth 1:10:34

Mm.

Uh.

Tarun Jain 1:10:41

Then you have similarity search and if you see you have similarity search with relevant scores which is your thresholding.

Then you also have similarity search with scores. There are so much of search techniques that one can explore, so I usually use maximum marginal relevance.

And I'll just add query and K equals to four. So does anyone recall what is K?

Mitesh Rathod 1:11:11

Uh, talking.

Tirth 1:11:12

The junk size.

Tarun Jain 1:11:14

Huh. So Villavanshank.

Ajay Patel 1:11:14

No.

Tirth 1:11:15

The translate.

Relations.

Mitesh Rathod 1:11:19

For relevant.

Tirth 1:11:21

Yeah.

Tarun Jain 1:11:22

So it's like if I ask a question, I have 10 chunk, but only give me 4 chunk which is making sense. Wait, 10 is very less, right? So I'll make this as two.

Tirth 1:11:29

Mm.

Tarun Jain 1:11:34

No results.

Of.

0.

OK.

I need page content.

Tirth 1:11:49

Hey, Cortana.

Tarun Jain 1:11:50

So.

Top player we are looking at 2 founders.

Tirth 1:11:55

This is a review.

Tarun Jain 1:12:02

If I use similarity search.

Tirth 1:12:06

I'm not getting anything out. It did something wrong. Maybe it's empty for me.

Mm.

Tarun Jain 1:12:10

It's empty.

Mitesh Rathod 1:12:14

I'm I'm getting, I'm getting three years. Same thing, top player as we are working with the two founders.

Even though I asked where is authentic located, I asked different question like what is the HR e-mail address of authentic?

RamKrishna Bhatt 1:12:35

I got something like contributing to open source is a powerful way to front end development like this.

Tarun Jain 1:12:45

Well, let's test if that chunk is there or not chunk of.

Ajay Patel 1:12:51
I'm getting a response, but with a whole thing like urgently can address and phone number, everything.

Tirth 1:12:52
Yeah, I know.

Tarun Jain 1:12:59
Yeah.
You're getting the response.

Mitesh Rathod 1:13:01
OK.

Tirth 1:13:01
I'm not getting any response. Why is my result empty?

Ajay Patel 1:13:01
Yeah.

Mitesh Rathod 1:13:06
You can tell the connection.

Tarun Jain 1:13:08
So can you process?

Tirth 1:13:11
Yeah, sorry.

Mitesh Rathod 1:13:13
Collection name changement.

Tarun Jain 1:13:13
Can it solve process if you give this correctly or not?

Tirth 1:13:17

I did chunk uh vector parents.

Tarun Jain 1:13:19

So you can you can check with this one embeddings not embed query. Any user query what is the length? You just have to copy that and paste it here.

Tirth 1:13:24

Yeah, yeah, that I did correctly.

Tarun Jain 1:13:31

OK, for me in chunk only it is not there the phone number.

Ajay Patel 1:13:37

And.

I get the phone number but it comes with other information also like site map partnership.

Tarun Jain 1:13:43

So now what is happening here is you're not using any LLM, so this is just knowledge transfer. So now what will happen is whatever context you got here will be going to the LLM.

You understood, yeah.

Mitesh Rathod 1:14:01

OK, I I am getting the address.

Ajay Patel 1:14:01

Thank you.

Tarun Jain 1:14:04

OK, even I'm getting it's there now.

Mitesh Rathod 1:14:04
Done with the others. Hope it makes sense.

Tarun Jain 1:14:07
The only thing is when you run this, you have to click on this three dot and here if you see you have the phone number, you have the contact ID.

Mitesh Rathod 1:14:14
Yeah, yeah.

Tarun Jain 1:14:16
So now what will be the flow? Let me open the slides.
We ask the query, we convert into embeddings. We look into vector database. We have the retrieve context. We are here now.

Ajay Patel 1:14:39
Mm.

Tarun Jain 1:14:39
So now we just have to add LLM and final response.

Ajay Patel 1:14:44
Mhm.

Tarun Jain 1:14:44
So if we do this two-part then it is rag. I mean huh that will be our complete workflow. As of now we are just experimenting with search which is our vector database.

Tirth 1:14:58
What did I do wrong? Can you help me? OK, what?

Tarun Jain 1:14:58

Did everyone want?

Can you share your screen?

Tirth 1:15:04

Yeah.

Tarun Jain 1:15:05

Guys, everyone got the results. You just have to expand it. If you expand it, you'll have. Since it was at the bottom, I was not able to see.

Ajay Patel 1:15:07

Yeah, I got the result.

Mitesh Rathod 1:15:08

Yeah.

Tarun Jain 1:15:15

If not, it's better to do print. It will remove the unwanted.

Tirth 1:15:15

OK.

Quadrant client collection name V3 client with path as DB2 with this loader.

Mitesh Rathod 1:15:23

Several block.

Tarun Jain 1:15:27

Oh, one second.

OK, can you Scroll down?

Mitesh Rathod 1:15:31

OK.

Tirth 1:15:33

Yeah.

Mitesh Rathod 1:15:34

Translate 1024.

Tirth 1:15:36

Is it 2048?

Mitesh Rathod 1:15:38

Yeah, about zero to 40.

Tarun Jain 1:15:39

No, no, that is fine. That is that won't cause an issue. Can you Scroll down?

Tirth 1:15:45

That is 762.

Tarun Jain 1:15:45

So this 768 it is correct, right?

Tirth 1:15:48

Yeah, 768 is correct. I tested it over here. Length it's 768.

Tarun Jain 1:15:53

OK, OK. Can you scroll down?

Oh, vector.

Can you run this again at the vector store?

OK, usually it should not take 0 seconds. There is. Wait, wait, it took 0 seconds. It should take at least 10 to 19 seconds.

Tirth 1:16:17

The vector store. Uh, where are the embeddings chunks?

Mitesh Rathod 1:16:19

You haven't added, uh, add documents. You haven't uh that.

Tarun Jain 1:16:22

Ha ha. We didn't add documents. Can you come to the new line vector store?

Tirth 1:16:26

OK.

Tarun Jain 1:16:30

dot add documents.

Ajay Patel 1:16:30

Add documents chunks.

Tarun Jain 1:16:33

Thanks.

Mitesh Rathod 1:16:36

OK.

Tirth 1:16:38

OK.

Ajay Patel 1:16:38

No, no. This will take at least eight to 101010 seconds.

Tirth 1:16:41

No, it isn't.

I see. I see. OK.

Tarun Jain 1:16:44

But what is that query? That query is just.

Tirth 1:16:47

And nothing was coming up. No, it was just so.

Ajay Patel 1:16:47

Very attempted.

Tarun Jain 1:16:59

OK, so we'll wind up here. So we start with query and we have vendor till retrieve context. Tomorrow we'll have the complete RAG pipeline and what we will do is we will replace in memory with cloud and we will use the actual data set that we want to use.

Ajay Patel 1:17:05

Mm-hmm.
OK.

Tirth 1:17:14

OK.

Tarun Jain 1:17:14

Which is the PDF that you guys were using yesterday, I mean two days back.

Ajay Patel 1:17:15

OK.

Tirth 1:17:20

OK.

Tarun Jain 1:17:20

So if you want to build chatbot for any specific data and if you have that within your what you call folder, even that is fine. And I remember one of you wanted to build Bhagwad Gita GPT as well. You can also upload Bhagwad Gita document.

Tirth 1:17:32

M.

Tarun Jain 1:17:35

But the only thing is the quality we have to see. This is basic rag, but we'll try to improvise on that particular data set itself.

Tirth 1:17:44
OK.

Mitesh Rathod 1:17:45
Tarun, just one short question. When for like what do we when do we select a line chain over line graph?

TJ **Tarun Jain** 1:17:51
Yeah.
Uh, there is no specific question. Line and mix is Langraf. It's blindly Langraf.

Mitesh Rathod 1:18:00
Can you?
Ajay Kumar.

TJ **Tarun Jain** 1:18:07
So there is no comparison between Langraff and Langchain. The comparison becomes Langchain, Langraff, Vexus, Lama index.

Mitesh Rathod 1:18:16
OK.

TJ **Tarun Jain** 1:18:17
This adds same ecosystem, so obviously we lose Langraf.

Mitesh Rathod 1:18:21
OK, but.

TJ **Tarun Jain** 1:18:22
There was also one interesting thing I probably teeth can drop off if in case there is another meeting. There is just one short thing I want to share. That's it. Do you guys remember yesterday we used prompt template?

Mitesh Rathod 1:18:29

OK.

Tirth 1:18:30

Yeah.

Mitesh Rathod 1:18:36

Yes.

Tarun Jain 1:18:37

I told we should not use prompt template right? Then I just was checking this Pinterest query book.

And these guys have used prompt template every single where even they're using line chain to be specific. So here if I just search for line chain.

So they're using open AI. If you see from linechain dot prompts they have prompt template, prompt template everywhere they're using only linechain.

Mitesh Rathod 1:19:10

OK.

Tarun Jain 1:19:10

Prompt template, prompt template. Then again one more prompt.

Mitesh Rathod 1:19:12

Yes.

Tarun Jain 1:19:18

And for open AI, they're using langchain open AI that open AI.

Mitesh Rathod 1:19:18

Oh.

Thank you.

Tarun Jain 1:19:24

But here you have one more use case which is using langchain.

Mitesh Rathod 1:19:28

Right.

Tarun Jain 1:19:29

But langen mixes langraf. That is not a comparison. Blindly you can use langraf. Langraf only comes at the end.

Mitesh Rathod 1:19:33

So.

Like uh.

OK.

OK.

Tarun Jain 1:19:46

It's a combination.

Mitesh Rathod 1:19:48

OK.

Bye.

Tarun Jain 1:19:53

So we saw Uber was using Blanchain, then we have Pinterest, then we have LinkedIn. Once I keep exploring, I'll tell a few more names.

Mitesh Rathod 1:20:02

OK, because I like when I'm from that, you know I'm working on that invoice thing, right? And I have to use AWS specifically and in AWS I'm using something called step functions and when I'm implementing something.

Tarun Jain 1:20:10

Bye.

M.

Mitesh Rathod 1:20:22

With Langraph I it is like I cannot make way for implementing that, but for langen it seems very easy. So that's why I was asking.

Tarun Jain 1:20:33

No, no. So land graph will only come at the end to manage our states. That's it.

Mitesh Rathod 1:20:36

OK.

OK.

Tarun Jain 1:20:41

I guess even GoDaddy's using Lanchen somewhere.

Mitesh Rathod 1:20:41

Thank you.

Tarun Jain 1:20:49

Oh.

Oh.

So this is the use case of GoDaddy, how they build their category generation system at scale.

Mitesh Rathod 1:20:56

Mm.

Tarun Jain 1:21:03

I didn't remember this.

They're using parenting output process which we used already. You will see this syntax.

Mitesh Rathod 1:21:13

Hmm.

Tarun Jain 1:21:16

Huh. But lunch and has very good, uh, parsers.

Even in Pinterest article if you read.

So what we use like the parsers, same things has been used here as well. So if you see line change parser, Jason parsing and the one here also it is parser only.

Mitesh Rathod 1:21:38

M.

That.

Tarun Jain 1:21:43

Which is 5 identical parser exactly same file we had used.

OK, yeah.

Mainly, that's it.

Mitesh Rathod 1:21:56

OK. Thank you. Thank you.

Tarun Jain 1:21:58

Yeah.

Ajay Patel 1:21:59

Thank you, Don.

Tarun Jain 1:22:01

Yeah. Thanks.

RamKrishna Bhatt 1:22:01

Thank you.

Mitesh Rathod 1:22:02

Set up.

◉ **Ajay Patel** stopped transcription