session we we can skip few slides. Uh so in this session uh we will focus uh
0:08
again we will reiterate a bit about the rug uh and uh we will go a bit more
0:14
deeper in embeddings vectorization and other uh rug uh systems and
0:21
possibilities and we should come uh to the state where we can even uh build a
0:28
quite big and reliable uh rock system that will utilize device rack at least
0:34
this slide I can skip uh so we will focus on the foundation uh then more uh
0:40
advanced rack patterns and uh some dos and don'ts uh for uh production
0:47
readiness and here we will cover uh even one of the question that uh we had for
0:52
the previous session so on the foundation uh so what is uh in general
0:58
like embeddings uh so we have uh the problem with just that examples like
1:04
words machine learning algorithms ML techniques and deep neural networks uh
1:11
that's basically computer doesn't understand similarity of that words uh
1:16
computers still understand only like numbers
1:20
uh so it cannot help us uh to to define just from the uh from the simple text
1:27
are they similar uh or ho how it to find so basic basically like embeddings uh
1:34
it's a trans transformation uh of the text uh to the numeric
1:41
representation on the vector. So when we talk about like machine learning uh then
1:46
for the for our embeddings uh we have like this representation uh of the
1:54
numbers uh depends on the uh dimensions. So uh it's like uh on
2:01
this uh exactly like image we have uh 536
2:06
dimensions uh so it's basically like uh graphs with the different points uh and
2:14
in a dimension uh of this size we we are putting um the the representation of
2:22
this uh word on that graph the same like for the ML algorithm pizza and stuff
2:28
like that. uh and uh if we are asking like uh about like machine learning for
2:35
example uh then computer based on that uh vector representation and number
2:43
representation uh can understand if like machine
2:47
learning uh for example because we we already um
2:52
uh the transform the representation in into the numbers if it's if it's close
2:58
to some of the like information that that we have right now in the vector

3:03
system uh or it's far. So based on like close and far uh it's basically like uh
3:12
ve vector uh databases when we set up like choose top five results
3:18
um providing to us with the answer uh from the five uh closest uh objects
3:26
uh and uh then retransform this to the text and we are basically getting
3:33
uh that's uh that text uh that's most closest to
3:41
uh to the text from uh our request in this case like for the machine learning
3:45
ML algorithms uh is quite close because the distance uh 0.02 02 uh and if we
3:53
just want to get only one like top element uh then we are getting ML
3:58
algorithms uh pizza recipes uh is quite hard so most probably uh system will not
4:05
uh suggest this to us um and uh if we uh are going a bit like
4:12
deeper to the exact like similarity search uh how it works uh so we uh have
4:19
our documents we already discussed about the rock pipeline. So uh it's uh already
4:25
uh transformed to the numerical representation uh from the embedded
4:30
models and it stores somewhere here to the vector uh to the vector store uh and
4:37
we have like three documents uh and we fully indexed that documents and it's
4:44
stored uh and when we have a search uh and if
4:49
we have like user query learn pipe Python. Uh then based on that uh
4:56
documents that we stored uh our system identify the distance uh for our request
5:04
and in this case quite close somewhat close and far away. If we set up that we
5:12
want to pick up uh two top results. Uh so in this case we will get uh Python
5:18
tutorial and Java programming uh as well. and then LLM uh maybe uh make a
5:26
decision that's uh to show only like Python tutorial but it can be the case
5:30
that it will provide as a uh output to the user uh both like Python tutorial
5:36
and Java programming because uh quality of the LLM
5:41
uh play uh quite significant role in this case as well uh quite often uh like
5:48
similarity metrics uh that uh systems identify. This is
5:54
angle between the actors uh vectors sorry uh straight line just distance
6:00
between them uh and uh in some cases some dot multiplication
6:07

um in that systems. Uh so why is like uh embedding quality
6:15
matters? uh because uh basically uh it's mean that uh how reliable answer uh
6:27
our user of our system will get uh and um right now uh MTEB
6:36
uh metrics I to be honest I don't I don't remember uh exact like u uh this
6:43
uh full abbreviation but I I I I provided the the link to the leader
6:48
leaderboard and you will see uh what exactly it mean what is the
6:54
abbreviation. Uh so this is the uh approach uh how to evol how uh current
7:01
embedding models uh evaluated on the quality of their basically embeddings
7:07
and accuracy of uh their embeddings. And uh in the in the point of time uh when
7:15
uh I was uh preparing this uh presentation uh few of the main or like
7:22
uh top embedding models basically from the Google and open AI uh and one of the
7:28
open source uh Nova search but right now uh I guess uh Chinese uh embedding
7:34
models uh on on a good spot there as well. Uh so you can try and play uh and
7:42
uh why the exactly like embedded uh quality matters. Uh if we will see the
7:49
uh different case studies or researches uh you will still see that uh any
7:57
embedding model or embedding system uh can provide uh 100%age of the accuracy
8:04
uh because uh still LLM or like similarity search can be like mistaken
8:13
uh mistakenly in interpreter uh and one of the like best result that
8:20
you can find uh like on the Google uh website uh for example uh it's research
8:26
about the legal discovery uh so they put uh 1.4 for uh million of different flow
8:34
documents um with utilization of Gemini embedding and they achieved 87
8:43
percentage accuracy and it's like for sure quite good number
8:48
um and uh with the another like application mind uh they achieved 82%age
8:59
uh of uh three top results recall uh in terms of uh the matrix uh it's as well
9:06
quite good result and uh in which uh what is good uh with the direction of
9:13
embedding models and uh we discussed a bit uh when we discuss about chunking uh
9:21
strategies that's when you utilize like additional ML or uh genai or embedding
9:27
models uh it's additional time for indexing
9:31
uh and it's uh additional time for the compute so it's much slower um

9:39
Gemini embedding uh already showing like quite good results uh in terms of the
9:46
speed uh so they tried to vectorize uh 100 emails on one of the study and uh
9:54
they achieved 21 uh 21 and a half uh seconds uh for vectorization of all of
10:02
that uh emails. And just to give you perspective of uh how it fast and what
10:10
is the uh how fast we are moving to improving uh all of uh these tools and
10:19
systems. Uh the previous result uh was more than 200 seconds. Uh so uh the
10:26
order of magnitude in terms of the uh speed increase uh for the
10:31
electctorization in 10 times uh so that system
10:37
much more like improved uh and we are going like on on that speed and that's
10:44
uh changes with uh many models systems and tools uh in
10:52
AI world because uh people experimenting um companies uh putting a lot of um
11:00
effort uh in um in this race so to say uh and uh if we will talk a bit more
11:14
about the databases um exactly for the vectorization
11:19
uh this is where I I wanted to mention about the speed of changes this uh as
11:24
well uh because u earlier I had this uh like slide why uh we need to have like a
11:33
separate database uh for embeddings and uh that's like uh current databases like
11:42
uh systems uh they have uh different algorithms uh they they don't have uh
11:48
everything that is needed for the embeddings that they quite slow for the
11:52
embeddings things uh and it was uh but uh postgree is moving with their um
12:01
with their extension they're moving in quite rapidly and catching up uh the
12:08
speed question and especially uh amount of the uh operations uh basically like
12:15
queries per second that they can handle and a lot of the
12:20
researchers already show that uh posgress is quite good alternative uh to
12:27
the uh specialized even um databases for the rock. Uh so you can see that uh for
12:37
example like uh faster than pine cone uh and for sure like cheaper and still open
12:44
source and um uh for for quadrant uh for quadrant uh it can handle much more uh
12:54
request per second but it's on a huge amount of the data. So they uh tested
13:00
this on the 50 million embeddings uh and this is where like quadrants start
13:07

degrading uh with the with the speed uh of working. So
13:14
potentially you can use just uh that database that you use uh that you used
13:18
to uh and you just uh need to add uh additional extension uh for the
13:25
embeddings uh and you can continue continue playing uh with your lovely
13:30
posgress but still uh we uh we have the databases
13:37
uh for the vectorzation and uh they still play uh their own game and for
13:44
what use case for for what uh systems they uh they they have the best results.
13:54
Um and it was already mentioned uh about like chroma today and chroma is uh quite
14:01
good when you are starting some of the prototyping
14:05
uh it's less rare uh used uh in um uh some big production uh systems uh but
14:13
they uh already have their cloud chroma I didn't try uh and still they uh have
14:20
uh some kind of like limitations um So chroma database are quite good when you
14:27
starting and prototyping because uh it's quite easy to start. You just pin
14:32
install chroma uh and almost everything uh is working. So minimal configuration
14:39
uh and uh you are starting almost uh like immediately with that. uh it still
14:46
has uh quite good additional functionality like for example meta
14:50
metadata uh filtering uh this is the additional information to your
14:56
embeddings uh that you can provide uh as example uh I've already mentioned uh for
15:02
example like category uh of the document uh that you are feeding
15:09
if you uh have like many of the document and chunks uh for example In the
15:16
metadata uh you can uh still additionally uh provide information
15:21
about what exact document uh from from which from from chunk is
15:29
uh and different like multiple uh collection support.
15:34
when we are moving already like to the stage of uh potential MVP or uh already
15:41
quite big production uh system uh then we need to have more reliable solution
15:47
and here we can look in the direction of quadrant uh or for example like posgress
15:53
with the uh PG vector uh extension uh and uh on the previous slide you you
16:01
saw that quadrant uh not so good on 50 million
16:05
um vectors uh in terms of queries per second. Uh but when it's uh in the

16:14
um in the measure between like uh 1 to 10 million uh it's really quite quite
16:21
fast and can handle uh a lot of the requests.
16:26
uh like for example for 1 million it's 626
16:30
uh queries per second and it's really quite fast and uh for for that if you
16:38
have um that range uh of the vectors uh and you need to have quiet rapid uh
16:47
embedding quadrant is a good choice uh in general
16:53
but if you have uh like billions uh of uh vectors then this is the
17:01
question about another type of the system and mil uh and they have uh this
17:09
embedding database in the cloud it calls uh it's already fully designed uh for
17:15
quite huge uh vector systems uh and it doesn't make sense uh to use this
17:23
database for like much smaller uh systems if you have under 10 million
17:28
vectors. Uh so do not recommend because um
17:36
that's uh that data and uh that uh systems that going from the left to
17:42
right uh it's then just harder uh like more complex uh to support set up and
17:52
configure. So that's why you are starting uh from the simp simplest one
17:57
just for the rapid start and when your system is growing uh you moving to the
18:03
let's say next one and uh each of them just fulfill their
18:09
purpose. Um and about like rock patterns uh rock
18:16
patterns in addition to the embeddings uh because we discussed about the
18:20
embeddings vectorization um it's not only uh one pattern or
18:28
approach um for the rock systems. Uh in addition
18:33
we can have uh this problem with the uh multihop reasoning uh with the
18:39
connection and uh in in the cases uh where we need to where we have like a
18:46
bit more vague uh request and we need to identify the connection for that uh
18:53
request. In most cases uh rock traditional vector rock uh systems they
19:01
are failing. Um and in example that that I provided and the problem that uh we
19:09
need to have like connection uh built from our request that uh user asked like
19:17
marketing, budget, compliance, GDPR and basically Europe. And this is the uh the
19:22

connections uh and this is exactly when we need uh already to utilize graph
19:29
system and uh graph rock uh in addition and it helps to increase uh reliability
19:37
uh of our system uh together with LLM and
19:43
quite quite often this is like a representation of the graph that we have
19:48
and when we when our user making the request. It's just like uh getting uh
19:54
some of the information from from the request and learning the different
19:59
connections and based on the uh connection that that we have in the rock
20:04
representation uh it can uh provide much better answer if we uh do not have like
20:13
explicit information and one chunk of the information that we are grabbing
20:18
from our vector system is not enough uh main players uh for the graph uh
20:25
solutions. So now forj uh falcon tiger graph map and rango
20:32
uh mainly uh we are playing like with the 4j because you can easily like
20:39
install it on your laptop uh and I guess it's one of the most popular solution I
20:46
would say. Uh the next uh problem that's uh quite
20:53
often uh come in the PD PDF processing problem. Uh so uh in a PDF uh we can
21:02
have um images in a PDF we can have tables
21:08
uh and maybe some formulas. So it's quite
21:16
uh complex um documents uh to to parse for the LLM and in the traditional uh
21:24
rock pipelines we have like OCR uh objects uh so basically uh the ML system
21:33
or it can be like LLM uh that uh can look exactly on the PDF uh and then
21:41
provide in the text uh in the text uh view describe what it sees on the PDF
21:49
and then we can just like vectorize it quite easily.
21:53
uh still we have like uh issue with the uh
21:58
tables uh quite quite often and here uh like some OCR again can help some other
22:06
approaches captions uh so instead of like using complex retrieval system uh
22:13
and that's relying on OCR uh because uh they quite often like failing with that
22:20
uh we can just use some embedding model uh that can understand uh what it see.
22:28
So just embed the image and one of the interesting uh approach uh to solve this
22:37
uh question for the last time uh it's quite like new let's say uh approach uh

22:45
it's called poly uh library uh changes uh so they instead of uh like using OCR
22:55
just um for the describing what information exists on the PDF and then
23:01
uh vectorize that information. uh they have like vision language model uh for
23:08
that and uh image representation just uh split
23:14
uh to the patches uh and then that patches uh basically embed uh to the
23:21
model and when we have the retrieval uh then we utilize that embedding model
23:30
and still like visual visual language model to to give the answer uh what we
23:39
have from that images. And this approach uh shows u
23:45
much better uh results uh in terms of uh like uh rock system uh retrie retrieval
23:56
of the information uh from the system and uh it showed better result like
24:04
15%age approximately than uh standard to three
24:08
wall system with the OCR are uh that we have. So to answer uh on the question
24:14
quite often uh when we utilize the PDFs when we have a PDFs as a documents uh we
24:21
utilize the OCR uh for describing of that uh documents and then we store the
24:28
information. uh but this approach with the Colali uh showing quite interesting
24:34
results and maybe it's something that will be used much more often in the
24:40
future as the as a part of this type of rack systems but still mostly uh OCR
24:46
approach used um and a bit about like aentic uh rug uh
24:56
because in AI right now everything ch changed to the agentic rock and for sure
25:01
this is uh quite interesting and uh quite reliable approach. So in the
25:08
standard truck we have query uh and that query
25:14
go into the embedding model uh then we query the embedding vector databases uh
25:21
then vector databases provide to us like candidates like top candidates and then
25:27
our query so our request plus uh candidates that our vector database uh
25:34
provided goes to the LLM M llm uh then process uh all of that
25:42
information that we put uh and then provides to our user the answer. uh when
25:49
we have the agentic rugg uh we have type of like router agent uh llms
25:56
uh and then uh within the tool set uh that's available for that llms uh it's
26:02

choose where to uh to w that request so to the rock lab search some external
26:09
APIs or taking control of the over the world and then uh only providing the
26:15
output message uh here we will uh have a bit more
26:21
information on that. So aentic patterns and maybe like query routing request
26:26
some of the the composition uh and selfcorrection approaches in the agentic
26:32
rock uh system. So when the user make the requests uh what the latest on AI
26:39
regulation so LLM router uh detect like latest then we will utilize the tool uh
26:46
road to web search so that's why I said that like in general uh if we are saying
26:52
that we add just additional tool like web search yes it's to some extent rock
26:58
system uh and uh like when we talk about the
27:03
query the composition. So we have uh compare our Q3 to industry and predict
27:10
Q4 and uh first of all we have the the composition of that request that uh
27:16
agents our LLM bricks like our Q3 matrix then we need to find this uh information
27:23
in our internal database or uh ve vector system uh industry Q3 benchmarks
27:31
potentially we don't have uh this information it's
27:35
publicly available then web search and Q4 factors again goes to the internal DB
27:43
but again it can go to the web search as well and uh the third uh pattern it's
27:49
selfcorrection so we first have uh retrieval
27:54
uh then we uh getting uh the documents uh then our agent uh based on some
28:02
identified our like threshold uh measure this and if uh score is like
28:09
uh lower than our threshold then it can even like rewrite the query and then
28:14
basically like retry again uh this approach with the retrieval grade dogs
28:21
uh and uh it can be uh cycled few times uh this selfcorrection
28:29
uh and when it's uh it's to use well when we have like quiet uh complex
28:35
worries and uh in cases if everything else uh that we discussed earlier failed
28:43
uh and you need to have like more higher accuracy.
28:47
Uh if uh we have when we shouldn't use this uh it's for the simple retrievalss
28:54
and if uh our top uh quality attribute for our system is
29:01
latency because uh you understand that uh this uh agentic systems when we have

29:08
LLM uh and needs to increase the quality of
29:14
their results uh the time of these types of the request is growing.
29:21
Uh I guess
29:26
we will need maybe uh Exana I guess our time is
29:33
end. Yes we are bit out of time. uh if you
29:38
have a time you can continue and also colleagues if you have a little bit of
29:43
10 minutes we can uh continue this session.
29:50
Uh I will try to finish this like in five minutes and then we will have five
29:55
minutes uh for for the questions. I just will not stop uh quite deeply on each of
30:01
the slide uh just few words and what is the information important from that
30:07
um in addition to increase the quality of our rack systems uh and why I
30:13
mentioned earlier that uh you shouldn't rely only on the vector uh search uh in
30:19
addition uh you can utilize like best match search this is what BM25
30:26
uh it doesn't have semantic understanding but it's uh calculate the
30:31
uh number of uh words uh that it finds like uh the same words that it find in
30:39
the different documents and can provide uh based on that uh statistic
30:44
statistical calculation uh what the documents we should pick up and on the
30:49
researchers uh when we have like a hybrid system uh approach in terms of
30:55
the search vector plus this BM25 it increase accuracy for uh a bit more than
31:02
10 percentage and DCG3 uh this is like top three uh recalls uh
31:09
basically top three candidates uh that that we saw earlier on the diagram and
31:15
when we add the ranker uh in addition uh it's even increased uh to 37.2%age to
31:23
percentage uh ranking it's additional layer uh that we add into the system uh
31:29
and we are grabbing from our vector system or from our database a bit more
31:35
results. So we are we are grabbing instead of like five 10 uh candidates we
31:40
are grabbing 100 candidates and then our ranking system uh try to understand what
31:46
is uh the best candidates for us and then provide as a result to the LLM like
31:52
five or 10 or maybe three of them and production readiness uh so basically
32:02

what do don't uh hybrid hybrid search is the best. Uh
32:07
we discussed uh in addition about like uh agentic search we discuss discussed
32:13
about uh the graph uh and where it's uh the most useful and what databases uh
32:20
you can utilize. Uh so based on your case uh based on what you need to
32:25
achieve uh you can use any use any of that tools. Uh metadata fil filtering uh
32:34
it's quite helpful uh especially when you need to have like metadata in
32:41
general like uh storing of the metadata uh quite helpful especially when you
32:46
need to have uh the resources uh seated like for example when you make the
32:53
request and get uh not only uh some information from your documents uh in
33:00
addition the uh citations uh of from what document uh that that information
33:06
is. Uh
33:09
evaluation of your system uh embeddings, reindexing uh and uh
33:17
evaluation uh
33:21
of the system quite often because you are making the updates and you can
33:26
continue improving your system. Uh so don't it's like opposite uh of of
33:32
the do uh so just uh quite helpful for you to uh not forget what you should do
33:38
for the production systems and thank you.
33:45
So uh we have one question in our chat. Mhm.
33:49
Uh does embedding library I use for for my rock must match embedding library I
33:54
used for training my LLM? embedding library for training your LLM.
34:05
Uh can you maybe give a bit more details?
34:09
Yeah. So before training LLM as I understand I need to embed like all the
34:16
text and everything and do do the embedding. Yeah.
34:21
Uh when you have a rock system, you do not train your LLM like fine-tune your
34:27
LLM. Yeah, it's not about I mean like when I
34:31
trained my LLM, I used some embedding for example from from Facebook or
34:36
Google. Do I have to use the same embedding library for my rug? I mean uh
34:44
as I understand like to put something into you also need
34:50
to do embedding. Uh do you mean when you are storing the

34:55
documents to the vector database and when you are retrieving the documents
34:59
from the database do you need to use the same?
35:03
Yeah. when when when I generate the answer knowledge base.
35:07
Uhhuh. Okay. And the version and everything should be the same and
35:11
identical with like with Yes. Uh sometimes especially like from
35:16
one provider they uh compatible uh one and another uh but you need to
35:25
double check uh this uh information but general answer is yes.
35:32
Got you. Thanks. Other question we have uh metrics like
35:39
NGCG or recall require ground truth answers. Should they come from humans
35:51
this one uh you are prepar you are preparing uh in most cases. Yes. because
35:58
you are preparing the uh expected result uh on your request. Uh so yeah mainly
36:06
it's uh human prepared metrics.
36:13
Thank you Maxim. Uh colleagues other questions maybe you have
36:23
I have a question. You're welcome.
36:27
Uh thank you. Uh so thank you for the session and uh in one of the latest uh
36:32
slides you mentioned that we have to evaluate often and I think this is the
36:38
most complex task when it comes to building AI based solutions. So
36:42
my question is whether we'll have any session
36:46
uh any session that will explain how to build this evaluations for ax systems
36:52
for overall like agentic systems. So the question is about evaluations.
37:02
Uh I will double check uh maybe in the next
37:07
sessions uh that we will have on the rug. uh enterprise productized maybe we
37:14
will have uh the session on evaluation or we will discuss with uh our
37:23
colleagues that uh we should potentially like add this uh as a part of our
37:30
education because right now what what I see I don't see like exact evaluation
37:36
uh in in the session but potentially it's a part of some of the next
37:42
sessions. Okay, thank you.
37:45
Uh why I said not only uh because uh preparation of the data it's not so easy
37:54

as well without evaluation you just don't know
37:57
whether you did good job at preparing your data or not. So you can be building
38:02
a system for like weeks or months but if you do not have any metrics to check its
38:07
accuracy and performance then it was for nothing. So
38:13
as always uh human in the loop can save you from that but yes uh
38:21
depends on the scalability of your system and
38:26
uh all of that parameters. Yes, I I agree. Uh but in general, uh what can I
38:32
say? uh it's not so easy still uh question for the evaluation in in
38:41
general uh with within work uh with all of this uh LLM systems uh because they
38:47
are like um they are not like deterministic
38:52
systems uh and still you need to identify uh the proper way how you can
39:00
like even identify that tops three that system should recall. So different
39:08
approaches in most the in most the approaches uh like some another LLM
39:14
utilized if you are not utilizing the people to to prepare the data. Uh so
39:19
it's still not not uh finalized question even for
39:27
for in general for the AI industry I would say.