1:55

Alexey

**This week, we'll talk about search. We have a very special guest today, Atita. Atita is an expert in information retrieval, also known as search. She has contributed to projects like Apache on OpenNLP, and she advocates for user-centric approaches. She's very passionate about promoting diversity in tech through groups like Women of Search, and currently she is researching RAGs. We'll talk about what it is. For those who don't know what this is – it’s an abbreviation, and not like rags. Welcome to the interview.**

2:30

Atita

Thank you.

2:32

Alexey

**The questions for today's interview were prepared by Johanna Bayer. Thanks, Johanna, for your help. Let's start.**

# Atita’s background

2:38

Alexey

**Before we go into our main topic of search, let's start with your background. Can you tell us about your career journey so far?**

2:45

Atita

For sure. Thanks for having me. I think I would definitely like to say at least one line before we begin – I've always followed DataTalks.Club since the very beginning. I think it's an honor to be here on the live interview – it’s definitely a first for me. Because we were talking about my name earlier, I think that's where I'm going to start. I am not sure if a lot of people have already noticed that my name is actually a palindrome name, as well as my last name – it's a palindrome. And as I can…

3:19

Alexey

**Oh! Both of them.**

3:20

Atita

Both of them, yes. [chuckles]

3:21

Alexey

**Is it a coincidence?**

3:22

Atita

I think it was a pure coincidence, because I definitely checked with my mom, and she told me that it wasn't really intended that way. It's also very surprising that the meaning of my name is… I mean, it's a Hindi-driven name, because I come from India, and it is derived from the word “atit”, which means “past”. So anything or the events that are driven by the past events – and I think our listeners would be very, very smart enough, if they can guess what exactly is driven by the past events, because that's probably what my mainstream job is, as well.

Machine learning models use past data to predict future events. So I think it's a mere coincidence that my name definitely falls into the similar stream as my career. And I'm pretty thankful to my parents as well for naming me this way. That was a short background of my name, and I hope we can now say it correctly. It's Atita Arora. I have no connection with the Northern Lights. [chuckles] Just to clarify. About my…

4:38

Alexey

**It’s A-rora, right? Aurora borealis.**

4:42

Atita

Yeah, that's correct. A lot of people think that my first name is Aurora, because that's definitely the case in Europe. That's fine. But my first name is Atita. Arora is my last name. So about the career journey, as obviously you already have in the introduction. Yes, I started in 2008. It's been 15 years. And if we count the time before my Master's, it would actually be 2007. I would say that now, I'm feeling fortunate that I started with the information retrieval space pretty early on in my career.

But at that point in time, as an early-20s person, I definitely wasn't too happy with, like, “Why am I being pushed into this field?” Because I was campus hired by this company, which was really big into working on revolutionary products, and it was set up by a Stanford professor. I was feeling lucky that he chose me out of a batch of 96 folks. But when I joined the company, the first thing that they asked me to do was like, “We want to work on detecting the relationships between two entities.” I obviously realized over the course of the journey of my career, that this is called semantic web, which I started working on in 2007.

Definitely, a lot of people did not know about it. The application that I started working on was based on Solr and Lucene. Solr was pretty early on – like version 1.2 back then. And there was no Elasticsearch. People were still struggling to move from databases to Solr, and they were really pounding on [their head] about why they need to move away from databases and what is not present in databases and why they need to have a full text search engine. So I was born – or my career was born, between that struggle. While lots of my friends were working on Java JDK applications and .NET, I was battling with this beast called Semantic Web. It was definitely not a very pleasant time, to be honest. [chuckles] But now I feel good that it happened to me very early on.

6:52

Alexey

**It got traction later, right? [Atita agrees] I remember already in 2015, companies were using Elasticsearch, Solr… It was already [becoming mainstream].**

7:03

Atita

Yeah, correct. Even Big Data didn't really have the name Big Data back then. I mean, when I was working on… I think only in 2012 is when I really understood that, “Oh! This is exactly what I do. Is this called Big Data? Okay. That's kind of interesting.” So yeah, I think in India, as well, we don't really get a lot of traction either – if I was in one of the Western countries (the US or Europe) back then, I probably would have been able to work on it more. So yes, I think this was the initial part of the journey. I think back in 2010-11, is when I was approached by this training company that works in a similar way as how Coursera works and they asked me to develop the course on Solr. So I think it was happening more and more, but it kind of sidelined.

I also got to work on a lot of content management systems, because search is a very backbone functionality of any content management system. Definitely, people can put on the content, but the major challenge is finding the right content. This was where I also realized that matching what the person needs –how to match the right content with the right person – and it just grew more and more as I progressed in my career. So I think this definitely kind of tricked my brain a lot more than that, and that's when NLP happened. I understood that natural language definitely has a lot of prospects, so about 2012-ish is when I was first using NLP in my project, and I was very surprised…

8:47

Alexey

**What do you mean by “NLP”? Because I remember, for me, NLP also started with search. There is this book, called Introduction to Information Retrieval, I think one of the authors is Manning (that's the last name). The book was pretty nice to read and I was recommending this book to anyone who was interested in NLP – as a good introduction to NLP. That's why I'm wondering – to me, NLP and information retrieval always came together.**

# How NLP relates to search

9:18

Atita

Right. I think, as of now, I would say that it is a definitely interchangeable term. But I think search, in principle – as I said, people were moving from databases to a full text search engine – people were still kind of hung up in that “token search” mode. So it was really hard. I mean, people didn't really realize that they could actually have more than one phrase and they could still have search results. I think this is what natural language processing kind of enabled, and we realized that with the content management system, it became even more and more important. People want to sometimes do things like describe the kind of document that they are looking for, sometimes with the content, sometimes with a title – so it definitely boiled down to, “How do we match the right content with the right query?” This is kind of where it went.

My curiosity into the other factors, for example, “What kind of query should match what kind of content, based on different kinds of business KPIs?” Definitely, there were a lot of other factors that came into the equation now. That drove me towards going for my second Master's in strategic business management. I wanted to understand more about, “How are these business components deduced?” I think this got really interesting because I never took up management as my first career, because I was very attached to technology. I wanted to make things by myself and I think that was very interesting for me.

Moving on, I think I wouldn’t be overbearing, if I say that I got a chance to work with all my dream companies. I worked with Lucidworks, which was a pioneer in search consulting, and then OpenSource Connections two years ago as well. This is when I realized that search quality is more than matching the right content with the right user and providing a measurable aspect behind it as well. So it was definitely…

# Atita’s experience with Lucidworks and OpenSource Connections

11:29

Alexey

**Sorry for interrupting you. I really want to ask you about Lucidworks and OpenSource Connections because everyone who works with search knows the names of these two companies. Especially if you go to conferences like Berlin Buzzwords, you usually see a stand from Lucidworks. They're quite present there.**

11:52

Atita

Sure. They have their own conference as well, Solr Lucene Revolution, which was later called Activate, when AI really became a thing. I attended those as well, as an employee.

12:02

Alexey

**I'm just wondering, how did you manage to get into these companies? Is it because you… Did you need to do anything special, or was it kind of obvious where they said, “Okay, we see that you've been doing search for so long. Come work with us.”?**

12:20

Atita

Oh, it wasn't that simple, actually. [laughs] I mean, if I talk about Lucidworks, one of the support engineers was in my class – the course that I mentioned, the one that I designed and delivered for almost two and a half years. He was in my class, and he reached out to me, “Join this company. I feel like you would be a great fit because of the way you describe things. It would be amazing for our clients to know how things are really happening in the background.” It was a pretty long process, to be honest. It was something like eight long meetings, and they grilled me on every different aspect of Solr and Lucene in terms of performance, and the application level and everything. Finally, I got in. It definitely was a very accomplishing feeling.

13:07

Alexey

**So just having the credentials of teaching a course is not enough, because they really want to test that you know the ins and outs. [Atita agrees] Because they’re consultants, so they need you to know that. From what you said, it's very helpful to teach courses, because your students eventually join companies. They always remember their teachers and they can recommend them.**

13:33

Atita

I would be candid to tell you that when I started teaching the first course, in 2013 or 14, if I remember correctly… I mean, I started in 2008, and when I was teaching in 2014, it was really hard because there were people in my class who were pioneers of Java applications, and they were asking me really low-level questions about, “Okay, so you're describing faceting? Like, how exactly would it work? I mean, can you tell me the low level of how faceting really works in a disk or in RAM? What exactly do you mean by doc values?” “It's a columnar index,” is something that I cannot get away with. They used to grill me and I used to really be like, “Oh my god! This is really taking all my energy.” I used to prepare for almost two-three days before that three-hour class on the weekend.

I definitely think that really pushed me going too far with what I really needed and I was digging into the resources and definitely using Lucidworks for a lot of my content preparation. At that point in time, there was no Stack Overflow or popularity of such searches (or reading material on it). So I always looked up to Lucidworks for different content and I was like, “How is it that I could work with such an organization?” That definitely had all my dreams and I expected that I would work with this company. It was definitely a very accomplishing moment. I think it was similar for the OpenSource Connections as well. Because when I moved on, I realized that maybe achieving good search isn't enough, unless you can explain what “good” is. I mean, you need to describe “good” in the parameters of the business KPIs, and I think that's what I got to know from OpenSource Connections.

I also think I got a brilliant opportunity to contribute to a lot of open source tools and projects. I've been contributing to most of the projects that they have. They’re very welcoming as well, and I think they have a very structured approach to address the relevancy and the search quality aspect of the search. I think they are the people who coined the term “relevancy” and “irrelevancy” of engineers. And they have a lot of courses and loads of content around it. So again, great blog post, and I think I’m a contributor on that too. [chuckles] I am taking that legacy of what I learned from them and I'm trying to apply some of that to my new companies as well. I've started working with Qdrant recently as a Dev Relations Manager. Hopefully I could benefit from all my learnings in my new role. Because we're on the topic of search, I think one thing that stood out to me at Qdrant is that, if I have an existing search engine, I could still experiment with vectors – I definitely always say this out beforehand – don't be smitten by stuff because it looks cool. It has to be very much implied by your use case. Everyone doesn't need to use vectors, definitely. You need to have some sort of investigation into if your use cases fit for vectors. If yes, then…

# Atita’s experience with Qdrant and vector databases

17:01

Alexey

**Qdrant is a vector database, right?**

17:04

Atita

Yes, that's a vector database or a vector search engine, if we can say that. I think in the world of everything being SaaS-based, I think it's a pretty decent proposition to go for. They're pretty amazing, because I also worked on Rust in 2017. I basically have a lot of trust in Rust. [chuckles]

17:28

Alexey

**So it's written in Rust, right?**

17:31

Atita

That's correct. Yes.

17:31

Alexey

**We had a demo from Kacper. I think he's still at Qdrant, right? [Atita agrees] He gave a demo maybe a year ago. I asked him, “Okay, it's like Elasticsearch but for vectors?” And he was like “Yeah. And written in Rust.”**

17:50

Atita

I would agree, actually. That was also my first occurrence when I started using Qdrant. I mean, the API's and the console are very driven by Elasticsearch – I'll be honest. If people love Elasticsearch, they will definitely love Qdrant – only that it's more scalable. And because I have always been the advocate of open source – all my previous projects, be it Solr or Lucene, Elasticsearch, or OpenSearch even, because that was the last project that I was working on – I've been the advocate of open source. Sometimes, because we used to work with the clients, people used to be confused as to, “Should I be using vectors inside Solr?” I think if we talk about putting vectors in your existing system, it definitely comes along with a lot of pain in terms of having to make changes into your config, and then iterations for ingesting your index items, and then obviously, configuring a wrapper around how these results would be served.

It definitely all starts with what exactly you want to achieve with vectors. So if that is what you want to do, I would recommend that you check out Qdrant, because you don't need to touch your existing search engine, because it can stand parallel to any existing search engine. It can also natively provide firsthand support for vectors only. The nature of being scalable and very, kind of, plug-and-play for vectors – it doesn't really ask you to bring your text search into the stack. You can keep using your text search, whichever stack you prefer, and for vectors, use Qdrant in parallel. That's definitely one of the perks – your existing application isn't really disturbed. You don't need to process everything else, and you can already kind of bifurcate from your messaging queue and put your vectors into Qdrant.

19:57

Alexey

**So I'm just wondering – I know that there is support for vector search in Postgres, for example. Then there is support for vector search in Elasticsearch.**

20:08

Atita

Yeah. Also Solr, also OpenSearch – all of them have support for vectors.

20:12

Alexey

**Lucene supports that and then automatically does all that…**

20:15

Atita

Correct. It fans out to all the products that use Lucene.

20:20

Alexey

**In the end, you can also have a standalone vector database.**

20:26

Atita

Exactly.

# Utilizing vector search

20:27

Alexey

**Let's say we already have an existing Solr installation, it's one of the latest versions, and it supports vector search. When do we need to go with Solr or Elasticsearch or when do we need to book for a standalone database?**

20:44

Atita

I think that's a good question. I've given several talks about using vectors inside Solr, Elasticsearch before. OpenSearch supports vectors natively as well. That talk was basically oriented behind people needing to dump their existing systems and move to one of these vector search engines or vector databases. That was not really the case. I mean, in your case, for example, if you can afford to have reindexing...

Because I've worked on some projects where reindexing is not really an option – there are businesses who cannot really reindex their dataset. I think this is a very classic use case where you could use Qdrant, because you're not really disturbing anything that's up and running – there would be no downtime for existing customers, they can still keep on using your existing solution – whereas for the experimentation with vectors and trying to figure out if that is what’s going to work for your solution, you need vectors. I think this is where Qdrant comes into the picture. I think there's a new blog post released today – Andre probably posted that on LinkedIn. I think it's a very apt description about when Qdrant comes into the picture and when it is not relevant.

Definitely, if you're looking for ease of operations and you cannot reindex, I think Qdrant is one of the solutions to go for. I loved that it’s not even pushing customers to things like, “We can do tech search for you. We can do this for you. We can train models for you. We can do N number of things for you.” We just do vectors. They’re very focused, and in my understanding, when you focus on one thing – because I have been focused on one thing for the last 15 years – I would say you turn out really good. You're not really distracted by so many different things – you’re not distracted by supporting multiple languages, text search or so on, so forth. So I would say that's definitely worth trying.

# Major changes to search Atita has noticed throughout her career

23:00

Alexey

**As somebody who has worked for 15 years in this area, you probably started with creating indices for Lucene in something similar to MapReduce without Hadoop in there. Now it has changed significantly since then. So now we’re talking about LLMs, vector databases. I'm just wondering, for you, in these 15 years, what were the major things that happened that you saw?**

23:30

Atita

Oh, a lot. A lot. I mean, it sometimes feels nostalgic. Sometimes, if I look back at the stuff that I did at my first company, it feels like I probably did that in my previous life because it's so different now. Initially, my challenge was – I was working with Solr 1.2 and Lucene 1.7, I think. They were both different packages. Making sure they work together was a heck of a thing, because it literally drained me with, “What kind of configuration would I put in that these two start talking to each other and doing the stuff that I really need to do?” From there on, they became one project in GitHub, and now they're separate again. From the feature point of view, if we take a look, after the introduction of more of natural language processing sorts of application features, the focus was definitely more on understanding the queries, and then synonyms and stemming and lemmatization and then promoted searches – all of these later came into the picture as well, as part of that era.

Then personalization was definitely another thing. Initially, personalization was limited to configured searchers or a configured brand inside the configuration. They were still very stagnant – not changing unless someone changed them. From there on people wanted them to be changing every week, or based on the customer demand, based on what's popular (popularity index). Things were becoming more measured and then applied to driven searches. Then personalization transformed into, “What two items were sold together? We have to promote this brand. We have to have recommendations that consider, ‘If this person is interested in this product, what is it that we should recommend more of?’.” Obviously, machine learning came into the picture, then Learning to Rank became a thing, so the ranking and sorting was affected by that. Now we see that it's all about large language models. If I was to say something on that, we're definitely moving towards more and more rich features with every passing day.

Now with ChatGPT, which turned one year old now, I think things have changed even further, because everyone wants a chatbot or a search bot in their business. People don't want to be limited to token-based searches anymore, people are not satisfied with synonyms or rules or search management, per se. People want more. People want their search engine to talk like ChatGPT and be action-based and so many other things. I hope I was able to answer that and cover some of the things. I might have omitted…

26:43

Alexey

**It’s pretty difficult to kind of squeeze 15 years into five minutes, right? [chuckles]**

26:49

Atita

Yeah, that’s true.

26:50

Alexey

**But also, I remember, when word2vec appeared, I was at university. Everyone was like, “Oh! Have you seen that? King – Man + Woman = Queen!”**

27:09

Atita

Oh, yeah! That's a very classic example. Yes. It's one of the examples that I saw in almost all the presentations that I saw. Even now, I think I saw the very latest presentation, which was…

27:25

Alexey

**Back then it was like, “Okay, you have this bunch of vectors. And then you have this Gensim implementation.” How do you keep these vectors? How do you use it? And then you implemented locality sensitive hashing (LSH).**

27:38

Atita

Vectors, in principle, if you look at them – I mean, it's not something that appeared out of thin air recently. It's been a thing since the 1970s.

27:49

Alexey

**But they weren’t databases, right? It's so difficult to choose a database. Back then, you would think, “Okay, I have vectors. What do I do with them now?”**

27:59

Atita

Yeah. Right, exactly.

28:03

Alexey

**But yeah, I think they were around for quite some time. Right?**

28:07

Atita

Yeah, they've been around since… I mean, they're older than me, [chuckles] if I was to say. The concept is older than me. And I think maybe the limitation in terms of the infra was one of the things that kind of held them back. And now that we don't have any limitations in terms of infrastructure – we have GPUs, and whatnot – that has basically enabled and made them fast enough that they can be used in production today. That's definitely something that has enabled it. But the concept has been around. If you look at the inverted index, that is also a kind of vector – it's just that it's in the manner of zeros and ones. As far as index, that’s also a kind of vector, it's just that the dimensions are going to be dictated by the number of tokens in your index, but it's obviously kind of a vector. If you look at it, that's one way to picture it. I'm very photographic in terms of my imagination – it’s easier for me to think, “Okay, it's not a new thing that I'm doing. It's just that we’re putting emphasis on the vectors being generated through another model that has the understanding of these tokens, and how the context of these tokens together would dictate into a pattern or a vector.”

29:37

Alexey

**Yeah, my Master thesis was about search, too. I remember reading this paper about vector spaces, which was from the 70s, I think.**

29:48

Atita

Correct. Yeah.

29:50

Alexey

**If not earlier. I think it was the 70s. Then there was another paper from the 90s – from 1990 – about applying SVD to…**

30:06

Atita

Correct, yes. Yes, absolutely. I have also linked some of these papers to my presentations. I was also pretty researchy about, “How did this come into existence? What was the early research about it?” Because, I was definitely trying to put on my data scientist researchy hat on, like, “What more can I find out about it?” I mean, I would completely agree. Yeah.

# RAG (Retrieval-Augmented Generation)

30:38

Alexey

**But now we have these things like RAG that, that I mentioned at the beginning, when I was reading your bio. Maybe you can tell us what this is? What is RAG? Why do you care about it?**

30:51

Atita

Yeah. RAG is the abbreviated form RAG stands for Retrieval-Augmented Generation. As the name suggests, there are two important pieces in here – retrieval and generation. What we tried to do, as part of my previous project with the OpenSource Connections – we were working with a client, applying retrieval-augmented generation in production for a very big research company. The basic idea is, as I said before – everyone wants a ChatGPT assistant on their website now, retrieval-augmented generation kind of enables it. What it does is, if you're using plain LLM in your search – imagine your question being sent to an LLM, which has very limited knowledge based on the data it was trained on – it would respond and sometimes you will see that the responses are not correct. It would make up things and these are generally termed as “hallucinations,” or I would say, a more black-and-white term that this response is not correct. There is no basis to justify that this response was correct.

That's when the augmentation of your data with the existing LLM was born. What we do is in this case, we are sending the query to the LLM, we provide the context with the client data. This was the augmentation of the context of the data. This basically helps the large language model, or, for example, the OpenAI model, to have the context of, “This is the question being asked. This is the context you need to answer this question with.” Then there are less chances of the answer being a hallucinated answer. So that is what I was working on. Because I've had a background of machine learning as well, one of the things was… We have measured everything, I think. In my opinion, this is kind of something that business folks also need – if there's something that you have worked on and cannot be measured, (it doesn't fit together) – I mean, you need to define the good. The good in this case was really hard, the evaluations of such systems are really hard because of the nature of the problem.

We need to have the evaluation to justify so many different things, like a go/no-go decision, because there's the reputation of the company at stake. There is the effort of the team at stake, and we don't want to spoil all of that. So we definitely need to have the evaluation to make sure that whatever we are shipping out is worth shipping out. Evaluation definitely covers that part. We also need to evaluate things based on the business KPIs, because now that we have access to KPIs (key performance indicators), we need to see that it matches with the business matrix as well. The evaluation of that system kind of becomes really hard because the main, underlying feature of LLMs is the diversity of the responses. It's not simple like search evaluation because we don't know what it will respond with.

One thing that I probably forgot to mention is that these 15 years, I've given that to ecommerce search, because that's where a lot of implementations and projects were coming up. I've had very rich ecommerce experience. I somehow love ecommerce, because the results are fast, you see the response and you see the customer adaptability to the new features very quickly. The response is very driven by certain query needs. What I was trying to say earlier was, in the case of ecommerce, we have a very clear structure in terms of, “This is the query. This is the response (or this is the response document). And this is how relevant this is.” So we had access to the trials, for example. We knew that this query would give a certain number of answers, there was a concept around precision and recall, but RAG is much more complicated. LLMs themselves need to give you diverse responses like a human being. So if you ask a question from a human being, there's a good chance that it would respond back in different words or in different ways to the questions. This is what we wanted with LLMs as well. And if something is not responding with fixed…

# Building a chatbot out of transcripts with LLMs

35:49

Alexey

**I just want to try to summarize what RAG is before we go into evaluation. This is an interesting topic. So, let's say… We have a podcast. There are 16 seasons, 9 episodes in each. It’s quite a lot of information, right? For each podcast, we'll have a transcript. And let's say that we want to build a chat application based on top of these transcripts. In Slack, we have a lot of questions about careers and other things. I'm sure for like 90%-95% of questions, there is an answer somewhere from one of the guests. It would be cool to build… That's actually a nice idea. Maybe we should do that.**

36:36

Atita

This is what I actually built. My God. I never open-sourced it. I open-sourced a very small RAG demo, but yeah… I am definitely cooking something up, which will be about RAG evaluations and also a RAG application that you hopefully could use. Yes, that also involves Whisper from OpenAI. That also involved podcasts. Not spilling beans about it, but yes – I was using transcripts and storing the vectors of this podcast transcript into the vector database. But the point being that again, another…

37:16

Alexey

**I'm just curious – with the podcast… So we can do it and then we can build a chatbot on top of that, right? [Atita agrees] But the problem with OpenAI – it has no idea about this podcast. Hopefully it does, but, if you ask a specific thing about search, for example, it might reply, but not necessarily with the information I want. I want to use, for example, our conversation right now. But it might just come up with some answer and we don't know if this answer is good, right?**

37:50

Atita

It's also timed. It is timed for the time when it was trained. There is a block – there is a cut-off for the information, so it doesn't know about any recent events. For example, earlier, we used to see with OpenAI that my response set is limited until, I think, June 2021 – something of that sort. So anything that has happened after it has to be explicitly provided. But if you provide that to OpenAI, or ChatGPT, for example, it gives marvelous responses.

# Ingesting the data and evaluating the results

38:24

Alexey

**So we need to find a way to actually tell ChatGPT (or GPT-3, 4, whatever), “Look, we have these podcast transcripts and this is the question from the user. How can we use the transcripts to answer the question?” Then our goal becomes (the problem we solve is), from all these transcripts, and each transcript has like 50 minutes of stuff there – how do we actually find the right thing? How do we find the answer to this question in this database? The answer is, we have a vectorizer and this is retrieval-augmented generation.**

39:06

Atita

Yeah, that's what I also figured out. There are different… I kind of wanted to save it for my blog, though – but it's four levels of the evaluation that we need in the RAG evaluation. What happens obviously depends on the model that you're using – the model has to be driven by, for example, how general it should be, or how domain-specific it should be. The second being, obviously, how the data has been ingested. There are several different statistics around it – how have you prepared the vectors for the podcast, for example?

39:49

Alexey

**How would we go about preparing it? You can ingest the entire article right, or you can just ingest every sentence, you can go with paragraphs – how do you select this?**

40:01

Atita

Correct. It depends on, again, the user experience. What do you want to deliver? If you don't want the user to be redirected to, “Oh, this is the podcast where you will find the answer.” That's definitely not the approach that businesses want – they want the exact thing to be given to the user, “This is where your answer is.” A very specific answer. This is where chunking comes into the picture, which is why you need to invest a little bit effort into how you ingest these podcasts transcripts.

This is where it comes into picture – the model that you've used, what the context length or the token length it supports is. Definitely, we also need to provide when we're going to chunk the entire podcast transcript – how much overlap should it have? For example, I may be referring to a lot of things that I might have introduced initially as bold, and then I will be referring to it/them/they. Obviously, the LLM would not really know what exactly “it” is, or what exactly “them” is? The overlap is of very, very big importance in that case. How much overlap does it need? We need to have an experimentation platform for that. Similarly, we need to have a chunking strategy, some of that obviously, Langchain has enabled – a small demo chatbot that you can build with Langchain. People have been discussing that all over LinkedIn if Langchain is production-ready, though.

41:30

Alexey

**What’s Langchain?**

41:32

Atita

Oh, Langchain is… Oh, that's kind of an interesting question. I'm not sure if you've not seen it, but Langchain is...

41:40

Alexey

**I’ve heard the name. But maybe some people haven’t.**

41:44

Atita

Oh, okay. I mean, from my limited view, and what I've used Langchain for, is that it provides you with different connectors in which your information can be retrieved, or information can be absorbed into your vector database or your search engine, for example. It also kind of sits between your retrieval and your generation stages in the architecture.

42:15

Alexey

**So retrieval is when we ask the bot something and then it needs to find the answer. This is the retrieval part, right?**

42:24

Atita

The retrieval part is more like… For example, if you visualize it – say there is a query, and there is a search engine in-between.

42:34

Alexey

**A query is what we put in the chatbot. Right?**

42:37

Atita

That's the question, yes.

42:37

Alexey

**For example, “I am a product analyst and I want to become a data scientist. What should I do?” That's the query. That's the question that the user puts in the chatbot.**

42:46

Atita

Correct, correct.

42:47

Alexey

**And what happens next with this?**

42:49

Atita

What happens next is, obviously, this query would either go directly against the language model, or, in our case, we would want this query to be sent to our vector search engine. What this will do is, your query is going to be converted into a vector, and your entire query is going to be converted into a vector search query. This is going to be sent to the vector search engine of your choice. This vector database would have the chunks, based on the chunk strategy, based on the length of the chunks that you've used, based on how much overlap you need, how many chunks you want to retrieve, based on the number of responses you want to have.

As you can see, there are a lot of different things that you can experiment with. As of now, we assume that we know whatever we want in this case, and we retrieve, for example, five chunks – five relevant pieces of information that would help ChatGPT to answer these questions. So what we do is – then comes our prompt – the prompt that you basically send to OpenAI, that you are probably a research analyst or research assistant. This is your context, and this is your query. Now can you answer this question and then you can define some guardrails, for example, “If you don't know the answer, please don't blabber. Please don't hallucinate. Just say that, ‘I don't know, I need more context to answer this question.’” And then you basically process an answer – a relevant answer, which is prepared by summarization of these chunks and is given to the end user. Obviously, you can also spice it up by providing references to the related documents where this answer was prepared from, which addresses the explainability of the response.

People want the AI responses to be explainable, so by attaching the resources, we could always say, “This is where the relevant information would be found.” Then there would be some sort of trust from the user that is built into the system in that case. But, as I was saying, evaluating such responses is very difficult because a user could have several different questions. It could be about a particular domain, it could not be about a particular domain. Obviously, we are not blessed that we would have access to the golden set or the judgment data. This is where the evaluation becomes really important. We need to break it down into multiple layers of evaluation. You could individually evaluate the model that you're using to generate the embeddings that go into the vector database. You could individually evaluate the chunking strategy and all the factors that are related to how the content is going to be split.

And then you can also evaluate the retrieval strategy – if four chunks are enough, or five chunks are enough, or we need ten, because the responses from the assistant are mostly, “I don't know, I need more context.” All that drives us into a more metric-driven system. Obviously, we can also have end-to-end evaluation, if the responses were correct or not, maybe using something like NPS (Net Promoter Score – thumbs up or thumbs down), which could be used to broadly identify if the system is addressing its purpose – if you address and answer the questions from the users. A thumbs up stands for, “I was happy with the answer,” thumbs down, “I was not happy with the answer.” There are ways in which we would want it to be adapted into a pipeline (a chatbot pipeline) or we could also do it in an offline way. I think my blog is probably going to cover both of these strategies.

46:51

Alexey

**I’m looking forward to that. I just want to go back to the example again about a chatbot for the podcast. Let's say we built that. We somehow split the article into chunks, then we ingest each chunk, we create a vector for this chunk, we store it in a database (Qdrant, for example), then the user comes with a question, then we turn this question again into a vector, and then we use the vector (query) to query the database. [Atita agrees] Then comes the retrieval part, which is a vector query, then there is the vector database.**

47:31

Atita

We're going to retrieve the content, yes. Based on the vector similarity, we are going to retrieve the content and this populates the context of the prompt.

47:40

Alexey

**So we include this in the prompt. So, “I am a data analyst, I want to become a data scientist.” This is the question, and then, “Answer this question using these chunks of content.”**

47:53

Atita

Yeah, “based on this context,” yes.

47:55

Alexey

**That's the “augmentation” part. Right? [Atita agrees] First there’s the “retrieval” – we retrieve it from the database – then we augment our prompt. And “generation” is – we send the entire thing to the LLM and then generates an answer.**

48:07

Atita

Right. That's correct.

# Keeping humans in the loop

48:09

Alexey

**And then we were talking about evaluation because, right now, I have this RAG system with all the podcasts transcripts, but now I want to see if it's working fine. I can, of course, go ahead and test it – make 3, 4, 5 queries and then see, “Okay, it kind of makes sense.” But there are so many moving parts, right? [Atita agrees] I can use different chunking strategies, I can use different LLMs, I can use different databases – the number of combinations of these parameters is just insane.**

**But I need to somehow come up with an OK thing that works, right? [Atita agrees] Here, we were talking about four levels of evaluation, at each level… but, for example, if I want to build the same thing, what's the easiest way to evaluate that this system is kind of working ? Do I use something like crowdsourcing?**

49:14

Atita

I think crowdsourcing could be one. It also depends on how big your team is. That's a major criterion. I was actually going to recommend a book like Human in the Loop because, ultimately, everything needs to go through a human. I think the systems aren't really well-equipped enough, although there are strategies in which it involves prompting to evaluate the responses as well. For example, what I found when I was preparing maybe 30 questions and the sample responses based on the different domains that my chatbot is going to be used for. All of these 30 questions would have one response. So what I could use is, as part of a more black-and-white matrix, the one that we prefer in search systems like precision recall, ECG, MRR, so on and so forth – we would use vector similarity like, “How much of my response was similar?” Again, we're using semantic similarity here. We can, again, use vectors for that. Based on these 30 questions, and 30 questions evaluated against my RAG pipeline, also they have fixed responses already given into the table, “How much of my response was similar to the response that I had set as the ‘good response’ to this question?” Again, we're still playing in the world of vectors.

50:52

Alexey

**So there's a book you mentioned, Human in the Loop. It's a book from Manning, right? [Atita agrees] It’s by Robert Monarch.**

51:01

Atita

Correct. That's the one, yes.

51:05

Alexey

**I know about this book. At DataTalk.Club, we have a thing called Book of the Week, where we invite authors to answer questions about the book. Back then Robert was busy working on the book and he told me, “Let's get in touch later when the book is published.” So partly, it has been published in July 2021. Quite a long time ago. [chuckles]**

51:32

Atita

[chuckles] Indeed. I was surprised when you said he was working on the book. Interesting.

51:39

Alexey

**It was some time ago, yeah. The community is like three years old. Maybe it's time we talked with Robert. Maybe we can invite him to the podcast, too.**

51:49

Atita

Sounds like a fab idea. Just as the rest of the podcast audience, I would definitely be on that one too. I would love to interact with him because I find this book to be really amazing.

52:02

Alexey

**Okay. Yeah. Making a note to contact him.**

# Application of vector databases for machine learning

52:07

Alexey

**I noticed that we have a question. The question is from Taras. Taras is asking, “Is there any application of vector databases for machine learning? For instance, could it be used for making the training of deep learning models faster? Maybe there are some other applications of databases for machine learning?”**

52:27

Atita

That's actually a good question, I would say, because there is a feature… I'm not sure how much of the space you follow, but recommendation is one thing that is a machine learning feature. Vector databases are pretty amazing at addressing this. Again, we can use vectors in many different forms – we can use personas as a vector. We could have the recommendation and have the personas, and maybe based on the persona similarity of two people, we could have a recommendation of the products. One of the other things that I saw is something that has been a recent change – recommendation itself has a very changed meaning as you were discussing the changes.

Now, people are more driven towards recommendation in a certain session, instead of storing the recommendations for each user, because a user is going to have several queries – 100 queries. So these recommendations have to be updated per session, and this is where vector databases, and especially Qdrant, (this is supposed to be launched today, or maybe in this week, there’s going to be one of the changes where we can build the recommendations based on every click that the person provides). This is where a vector database is going to be amazing, because these recommendations will be updated on the fly based on every single click of the user.

54:03

Alexey

**Yeah, that’s really cool. I know there’s this classical approach to recommender systems, which is called collaborative filtering. You take all the users, you take all the items, then you have this large matrix, and you somehow reduce the dimensionality of this. In the end, what you have is – vectors for users, vectors for items – and then what you can do is just take all the vectors for items, put them in a vector database, and when the user comes… or maybe you can pre-generate it, where for each user, you basically query the vector database and then you get recommenders. [Atita agrees] There are many different problems with this approach, right?**

54:48

Atita

Right. Re-ranking was another way in which vector databases could be of help.

# Collaborative filtering

54:54

Alexey

**What you mentioned is… With collaborative filtering, we would need to re-do the whole thing, right? Then the vectors we do from another training will be super different from the first training. What you mentioned right now with clicks updating for sessions – there are other techniques you can use?**

55:14

Atita

And it doesn't need any fine tuning, it doesn't need… In the case of the cross-encoder method for the ranking, I think you can completely skip that part and you can do it on the fly. That actually reminds me of something really funky from 2016. The first time when the recommendation was introduced by Amazon, I remember that there was a talk given by someone from Amazon, who apparently bought a toilet seat, and he wrote to Amazon that, “I had one toilet in my apartment. I bought one toilet seat. I don't need any more toilet seats. Just get this toilet seat recommendation out of my feed. I don't need any more toilet seats.” So I think the recommendations have also come a long way from then. I think it's session-based… For example, like TikTok users or any video platform users, the recommendations are basically built on the fly. That’s what the next big thing is.

56:08

Alexey

**This means, “Okay, I bought my toilet seat. Now I'm searching for something else. You can see that in my last sessions, I was checking out pencils.” Then you see, “Okay, I have checked out this pencil, that pencil. This person is probably interested in pencils. Let's just show him a ton of pencils instead of the toilet seat that he bought earlier.” Right?**

56:32

Atita

Right. That's very, I would say, the sensible approach.

56:39

Alexey

**I’m searching for pencils right now. That's what I'm focused on right now. So it makes sense to show me what I’m interested in.**

56:44

Atita

Context actually becomes the thing.

56:48

Alexey

**Yeah, that makes sense. Amazon is probably doing something smarter so they can know, now, the things recurrently ordered, things that you ordered and never want to order again, like a toilet seat. For example, I like peanut butter and I use Amazon for ordering that. They know that I might run out of peanut butter, so they’re like, “Hmm… How about you buy it again?” So they know when I run out of peanut butter, so they can push the recommendation to my face. And when I just bought peanut butter, I didn't see this recommendation – there is something else. They are very smart – they know when I will need it. They probably have a ton of different vector databases.**

57:31

Atita

For sure. There are a lot of different permutation combinations, which is why I love ecommerce. It's so happening all the time – there's something coming into existence. There's always a new user need. That really pumps me up about the field.

# Atita’s resource recommendations

57:50

Alexey

**If somebody wants to learn about classic information retrieval, before all these vectors, would you say that Introduction to Information Retrieval is a good starting point? Is there something else?**

58:01

Atita

I would say so. I would highly recommend you read the Relevant Search book as well. That's definitely one thing. We have tons of blogs and whatnot.

58:15

Alexey

**Relevant Search. Relevant Search covers… Is it about Elasticsearch or Solr? I don't remember. It's Elasticsearch, right?**

58:22

Atita

I think it's very search engine-agnostic in terms of the book, which is why I recommend it. There would be examples that come along with the book, hopefully also Solr-driven as well. Mostly Elasticsearch. But I think the idea is to communicate the thought process and knowledge of the concepts, which I think are delivered pretty well with the relevancy guy himself. You know who the relevancy guy is? [chuckles]

58:47

Alexey

**Duke…**

58:48

Atita

Doug Turnbull, yeah.

58:51

Alexey

**He gave a talk at DataTalks.Club at some point. It was pretty nice to have him, actually. When it comes to more recent developments, like all this RAG stuff, is there a good resource? You mentioned that you're working on a blog post? [Atita agrees] But is there something else you would recommend checking out if somebody wants to learn more about this stuff?**

59:18

Atita

From the evaluation point of view?

59:22

Alexey

**Just in general and also about RAGs?**

59:26

Atita

Oh, well… I think Langchain (the site itself) has plenty of different things that they offer – different ways in which RAG could be achieved with different search engines. I would definitely recommend reading about it. Plus, there's a new resource where I will be contributing my blog on Vector Hub. That's another place that I can recommend you checking out – great content. We're working on a lot more interesting content as well. Nothing else comes to my mind.

60:01

Alexey

**Relevant Search, Langchain documentation, then Vector Hub. Please send us the links. [Atita agrees] Whenever you publish something new, we're also interested to know that. I assume every time you publish something, you also make a post about that on LinkedIn, right? So we can follow you there.**

60:21

Atita

I try to do that.

60:24

Alexey

**Yes. So please make a post when you publish that evaluation article. With that, I think, that's all we have time for today. Atita, thanks a lot for joining us today.**

60:39

Atita

It was a pleasure.

60:41

Alexey

**Now I know that your name is a palindrome. It never occurred to me to actually think about that. [chuckles]**

60:50

Atita

Mission accomplished. [chuckles]

60:51

Alexey

**I wonder how many guests we had in the past with a palindrome name. I don't think we had any. I'll check that. [chuckles] It’s good that we had the first guest with a palindrome first and last name. I think that's all for today. Thanks, everyone, for joining us today. Thanks, Atita, for sharing your experience with us.**

61:08

Atita

Thank you so much for having me and thank you, everyone, who's been listening today. Thanks. Have a nice day. Bye.