01:40

Alexey

**Let's go. This week, we will talk about knowledge graphs and LLMs lamps and how they are used in research in academia and industry. And we have a very special guest today, Anahita. Anahita is originally from Iran. She transitioned from mechanical engineering and specialized in applied mechanics in Sweden, then worked in the automotive industry for five years.**

**Then she shifted focus to pursuing a PhD in applied AI in Germany. And she combined her love for life and learning with a passion for cognitive sciences. And she likes dancing, painting, and running. Yeah. This is a summary from ChatGPT.**

**I think it's doing a relatively good job in summarizing. As always, the questions for today's interview are prepared by Johanna Bayer. Thanks, Johanna, for your help. And welcome, Anahita, to our interview.**

2:48

Anahita

Hello. Thanks for the invite and help. Also, thanks for the intro. It was covering great.

# Anahita's Background

2:57

Alexey

**Before we go into our main topic of knowledge graphs, LLMs and all that, let's start with your background. I briefly mentioned your background, but you can probably go into a little bit more detail. Can you tell us about your career journey so far?**

3:16

Anahita

Yeah, it was always a bit tricky to know what the next step was. But, as a summary, you said it perfectly. I was studying mechanical engineering. It was… I think I really loved math, but I wanted to be practical. So that's how you end up being an engineer, I would say, basically. And then I moved to Sweden because I was really curious to study a bit more. Because in the Bachelor, I really felt that I didn't learn much. I didn't really feel [what it was like] to be an engineer. That's why I went into applied mechanics to really feel it – that you can use it in different analyses. And then I started working and living in Gutenberg.

It was a bit hard to be out of the automotive industry. Everything was quite random, I would say, with a bit of higher probability. Then, starting in the automotive industry, it's a lot of demand for automation. I was really lucky to be in this company because there was a lot of unestablished coding and automation. Usually you don't get this chance of doing a lot of coding in companies as a fresh graduate. Then a lot of code and also, the plan was that after a couple of years, I would pick the topic that is interesting for my PhD. I thought it would be two years, but after two years, I had no clue what PhD I wanted to do.

Then after four years, I figured “Okay, maybe I'm looking in the wrong domain because I spend most of my time in optimization, in automation, and a lot of semantic reporting for the company.” Then I dared to feel that “Okay, maybe I should move toward computer science and data science.” That's when I started to shape my PhD proposal. I had a tough time getting funding in Sweden and that's why it took me to Germany – to work at Fraunhofer SCAI on the topic that I’d like to do my PhD on. Of course, semantic reporting was the next step for that – to have a knowledge graph for crash simulations.

# Mechanical Engineering and Applied Mechanics

5:37

Alexey

**What do mechanical engineers actually do in applied mechanics?**

5:44

Anahita

The program is focused on finite element analysis and CFD. So it's like you break down the physics into small elements and then try to find the behavior of them. I mean, that’s a really basic explanation. You have a lot of different physics of which you want to predict the behavior, and then you try to use different questions to find out what the response to that change to a force is – to a vibration, to a flow of fluids, or to the formation of a crash as well.

6:29

Alexey

**I guess the application of this would be in the automotive industry, like the designing of a car, right?**

6:38

Anahita

Yeah, for sure. Before, in the past, it was making a lot of prototypes (physical prototypes) and doing tests. Since they really wanted to shorten the development, and also save a lot of money (because making a prototype is a lot more expensive than building a real car). They are one of the leading industries in using finite element analysis. For example, truck development, or any other developments – they are far behind compared to automotive, because of the high demand of these kinds of analysis.

7:17

Alexey

**So there’s a system with different moving parts, and you want to change one of the parts of the system – you want to predict the behavior after this change, right? And this is what you do?**

7:31

Anahita

it's usually not like… You come with the design and you say, “Okay, I want a car that looks like this.” And then you say, “I want to have these performances.” And then you try to see how we could fulfill all the requirements. Usually the surface (or the target of the market) is really defining how we should do the improvements. Usually, it comes with a targeted direction of the performance.

# Finite Element Analysis vs. Machine Learning

8:05

Alexey

**So this finite element analysis – is it in any way related to machine learning? Or is it a totally different thing?**

8:15

Anahita

We could say… It's not that you… Hmm… You could say it's still numerical analysis. It's not unrelated to differential equations [chuckles] but you're not solving it based on cost functions and data. So that's the different thing. You try to model the behavior and predict it. A lot of investment (a lot of focus) is on developing a good material model to do the prediction.

8:50

Alexey

**This is what you studied and then you said that there was no way for you to avoid the automotive industry. You were doing optimization and semantic reporting. What are these things? What kind of optimization were you doing? What is semantic reporting?**

# Optimization and Semantic Reporting

9:08

Anahita

So, with the semantic reporting, what I'm referring to is that they are… it's an interesting topic to have FAIR data – I mean to be able to regenerate the analysis – because what we are seeing ahead of all these tests and crash tests, is that you don't need to do any physical tests anymore. So it's kind of that on the release vehicle, maybe you release a model of finite elements and then all the tests will be done. I don't know if you know, for example, all of these tests, when a vehicle comes to the market, you really need to rate the vehicle or you need to pass some legal requirements to have it sellable on the market.

9:55

Alexey

**So the videos where they put mannequins inside a car and then car drives into a wall.**

10:04

Anahita

Yeah, a lot of different… Yeah, exactly. A lot of sensor measurements, looking at the injuries of the occupant and a driver. The main measure is actually the injuries. Exactly. And… I forgot the question. What was it?

10:22

Alexey

**I was asking about optimization and semantic reporting. Are they related? They’re different, right?**

10:28

Anahita

Yeah. From the reporting [side], [regarding] this point – we need to be able to re-evaluate or regenerate the results that we are reporting. For that aspect, also, within analysis, we generate a lot of simulations. For example, for one vehicle, there could be more than 500 simulations. With semantic reporting, the main focus was on reusing the data and, in the long run, to be able to regenerate the results. One crash simulation could have 12 million elements, and then it could take around 12–17 hours on like 192 CPUs. It's a really costly analysis.

With the semantics it was really [about] classifying all the measurements we have – like the sensor data, and barriers, and so on – to be able to compare the results. In the past, it was generating PowerPoint and Excel sheets, so there were a lot of scripts that were auto generating PowerPoints. But still, you really couldn't compare. With the comparison, I have seen two engineers putting two PowerPoints that are generated with scripts and trying to compare curves. So that's one of the real changes – you can click ‘analysis,’ and you can overlay all the sensor data. After that, within knowledge graphs – coming from the semantic reporting – is to find the patterns in those.

Within optimization, a lot of time within engineering, you try to change thicknesses or adding holes, and so on. So it was trying to use these geometrical changes to be optimized to the behavior we required. So it's a lot of topology optimization, and the designs. For example, for pedestrian analysis, one of the main components that is really important is the inner hood, because you hit the head there. It's about how to cut the mass and still have a good stiffness, but not so hard [that it’ll be bad] if you crash into a pedestrian. That's the target of optimization.

12:41

Alexey

**And all of that happens on a computer, right? [Anahita agrees] I assume you don't actually crash into pedestrians to see…**

12:50

Anahita

No, luckily not. [chuckles] But all of these data, we also try to model. Also, in the computer, we don't really hit a pedestrian – we have a head, or we have a leg impact and there are sensors in them to measure the accelerations to see if this is okay or not. Then, after a while, we still do a real physical test, but not on a real pedestrian, but on this model of head or leg to evaluate how good our models are.

13:21

Alexey

**I understand why it's called reporting, but why is it semantic reporting?**

13:25

Anahita

Because you define all these measurements. So it's a kind of modeling of data behind it. There's also a bit of semantics on generating the PowerPoints, but still, to be able to compare these simulations, you need to have semantics to connect different analyses together.

13:52

Alexey

**Yeah, I think I more or less have a picture in mind. There are different experiments. These experiments could be related. For example, there could be, I don't know… Can you maybe give an example of two experiments that are related? Could be something like: we are in a car and we want to hit the wall and see how much the driver is injured. But another experiment is how much the person who is sitting nearby is injured, right? So this could be two different experiments that are related?**

14:33

Anahita

Hmm… No, I mean, it's really sensitive analysis, I would say. With these relationships there are really much, much smaller changes. It could be that, for example… We make a lot of impact points on the hoods and to see how good the performance of the whole vehicle is. Usually for example, with a distance of 100 millimeters, so quite small still. Then, one of these relationships could be like, how much the area around it… For example, if slides a bit on the impact, how robust is that and how much are they still staying with the same behavior of the acceleration? Because acceleration curves are really sensitive.

It's also hard to model them – to get the correct acceleration curve. So I would say, from driver to passenger, it's a lot of change to say how much they are related because most vehicles are not really symmetric. You have an engine there, and you have the cooler there. We try to have a symmetric intrusion, but it's really tricky still, to have it. In real life, most of the impacts are not symmetric either.

# Application of Knowledge Graphs in Research

15:58

Alexey

**Do I understand correctly that this is something you started doing while still working in the automotive industry? Then this is what led you to do graphs now? [Anahita agrees] You were talking about these impact points that are close to each other and this means that you can build a graph of all these sensors or impact points somehow. Right? [Anahita agrees] This is how you arrive at a graph?**

16:31

Anahita

Yes, exactly. I mean, this is also like when I was trying to develop this web-based reporting (or semantic reporting) since there were many analyses, it was really old-fashioned old-web technology. Then, it was quite necessary to have a backend and a database to improve the poor performance and also frontend.

At that time, when we started to decide what kind of database we needed to use, we decided to skip relational databases. I would say, that's when looking into graph databases started because the automotive industry has so many changes that if you want to have a relational database, it's really time-consuming to maintain it. It's like when I started with my company – we were looking into implementing data management systems. This is an old topic within automotive – to maintain the part.

Because one vehicle has a lot of different parts, a lot of different models, so how to connect these? The input data, how to relate it? They are looking into relational databases, and to really establish it, it's time consuming. That's why my mentor advised me to use Neo4J or any other graph databases. That is where it started. But it's also not easy to just store crash simulations or any of the analyses in a graph database, because you can’t store all of them – all the data.

18:10

Alexey

**I'm trying to visualize this graph and I don't think I can. What are the nodes? What are the ages in this graph? What kind of relationships can we model there if we talk about semantic reporting?**

18:27

Anahita

Yeah. So one of the basic [places where you can] start is following the R&D development from a vehicle that is supposed to be on the market. Say we use crash [data] – because it could be any different analysis and requirements, like durability and so on. Within the crash [graph], we know what the year of the release of the vehicle is, and then what the required performance is. This is some kind of semantics to connect the current vehicle to the vehicles on the market and be able to compare the performance of the…

So it’s a kind of benchmarking to see how we could perform – from the weight of the vehicle, the size of the vehicle, and so on. And then, coming from vehicles, one vehicle that we had, it has a platform and also an upper body of how the vehicle looks. So it's like really capturing the structure of the company and how they are developing, because these relations also help to relay the analysis and the behaviors. So why, for example, is this platform’s upper body important? Because you could compare the sibling vehicles. You see the cars that look the same, but they have different ride height or a bit bigger trunk size and stuff? These vehicles could be related.

And it was a topic of interest to see if it could predict simulations based on one platform and upper body to another platform and upper body. If we go into a bit more detail, between simulation – it was to convert the physics of the problem to a graph. What is really important for crash simulation engineers to detect a behavior of a crash, to say, “This simulation is similar to another simulation from this analysis concept.”

# Graphs vs Tabular Data

20:32

Alexey

**While you were talking, I was taking notes, and then I drew a graph. Within the graph, I have a node with the year of when the car was made – let's say 2020. This is the year, and then we can have notes of individual cars and the connection between the car and 2020 could be the year when it's made. Then there’s the body type or upper body type. [Anahita agrees] And then I guess there are the different characteristics of a car, right? All of these are nodes and each car is also a node, and the relationship between the nodes (the ages) is something like ‘made in’, the type of the body, or engine type – different characteristics, different figures. [Anahita agrees]**

**Okay. And why is it a useful representation? Because if I think about machine learning – in machine learning, we quite often have tabular data, right? We can have rows, where each row would be an experiment, and then we can have different features. The features could be year, type of the body, upper body, type of the engine, other things. The result, (the target – the thing we want to predict) could be… I don't know, what do we usually predict? Something like impact on the person or on the pedestrian or whatever? Right? So why do we actually need graphs here? What can we not express with a table?**

22:11

Anahita

I mean, a lot of things that you have as a graph, you can express with a table, but then it's mostly an abstraction. That is nice. Also, to find a bit more complex relationships between them. For example, one of the main gains I have in my PhD is to have a visualization where you can have a look at over 300 simulations. So if you want to compare 300 simulations in a table with a visual look, I think it would be a bit tricky. With using weighted graphs, and visualizing the simulations, and then there are parts that have been involved mostly in the crash – It was simply clustering the simulations, and also showing what the important parts are for those across simulations. What are the common parts of those?

Also, for example, to detect a load path. When you really transfer the physics of the problem into the graph, you can also answer different questions like load path detection. Load path is… If you transfer a structure of the vehicle, the graph, and each part is one node, and then if they're connected, they have an edge between them – so connected to other parts – and then you try to get how much each part has been involved in the crash. Then you try to find what the main path of the load transfers within the structure – so the longest path used. That is also a highlight that I don't know of a way to really see it in a table.

23:59

Alexey

**So that's what you do in your PhD research, right?**

24:03

Anahita

Yeah, exactly.

24:05

Alexey

**As I understood… So you have a system based on Neo4J and then you have different graphs – you have a graph with different nodes, and then it allows you to do these explorations. You can click on the node – let's say we want to look at cars that were made in 2020 – then you can maybe click on the 2020 node and then explore different models and how they performed in experiments, right? [Anahita agrees] You see all these load paths and other things, right?**

24:41

Anahita

I would say some part of these knowledge graphs is in data engineering, and I think most of the things you can do within knowledge graphs (maybe I am not really that experienced to comment on it) but I feel you can also do it with relational databases. It’s just a lot of overhead of maintenance and so on between these two. But what is quite interesting for myself is when you can extract a computational graph from a knowledge graph, and then do further analysis on that. These [things] that I referred to are mostly on graph data science, specifically compared to knowledge graphs.

25:22

Alexey

**So a knowledge graph is… So we have a car and the parts of the car have different relationships between each other – they are connected. Then there are all these sensors – the different characteristics of a car, like year, type of upper body, all these things. So this is the knowledge graph, right?**

25:46

Anahita

Yes, exactly. And then you get the node degrees so you can do a query there. So what are the other cars related to this one, whether the parts and so on – so you can do a lot of queries on that level.

26:00

Alexey

**This is how we express what we know about the cars. But then we do this crash simulation and somehow also record the results in this graph. Or how does it work?**

26:13

Anahita

Yes, exactly.

# Computational graphs

26:15

Alexey

**And this is the computational graph that we have? I think I don't understand.**

26:21

Anahita

No, that's not. So we have the knowledge graph that includes a lot of simulations. It holds the market, different cars, but then it also has a lot of simulations of that vehicle included. Then we pick, for example, some simulations and parts as the input data, and then we make a graph (like a NetworkX graph or whatever) of graph data science. That is the one I'm calling the computation graph. That is not related to FE analysis directly anymore. So it's just graph data science, or machine learning.

27:01

Alexey

**Yeah, and what do you actually mean by “graph data science or machine learning”? What kind of things can you do with these graphs? You mentioned NetworkX, which is a library in Python that can work with graphs, right? [Anahita agrees] So once we express this graph on NetworkX, what can we do with this?**

27:16

Anahita

One [thing] is predicting the similarity of the simulations, like this longest path analysis, as I told you, or a lot of visualizations. When you load the whole Knowledge Graph to, for example, as a weighted graph, it's especially not easy to have weight on all the edges, then it will be a bit tricky to have a good visualization. So that's why I call it a computation – you always look at part of your whole knowledge graph to visualize it and find the pattern. But still, the whole data is coming from the knowledge graph.

# Graph Data Science and Graph Machine Learning

28:00

Alexey

**I'm curious. You mentioned two terms, “graph data science” and “graph machine learning”. If I understand correctly, the distinction here is… In one case, it's more like doing some sort of analytics, right? Or doing this analysis – exploring the graph. And then the other case is making predictions? What's the difference between these two?**

28:21

Anahita

I think graph data science and graph machine learning are the same. So it's like saying, what's the difference between data science and machine learning? Same difference, I would say. [chuckles]

28:31

Alexey

**[chuckles] Okay. So what kind of things can you…? Okay, you actually mentioned things that we can do, but I'm curious about when it comes to machine learning – when it comes to predicting something. We can see if two things are similar using different similarity metrics – they are more like graph similarity metrics. I don't remember. I remember that there are different measures…**

28:55

Anahita

SimRank is the method I use. There are items that are referred to by similar items that are similar. So that's how you predict the similarities of items. Because you really don't have any ranking for the simulation similarity. So you give as the input, for example, one simulation and rank ‘what are the related other analyses’ to that.

29:23

Alexey

**So it's more like unsupervised machine learning. Here you don't really have…**

29:27

Anahita

Exactly.

29:28

Alexey

**Okay, I see. And then, your task is to understand the similarity and you used different (in your case, it’s SimRank) to understand how similar two experiments (or two simulations) are. [Anahita agrees] Do you use any supervised machine learning?**

29:48

Anahita

A tried a part of it and that was really on a really, really small “toy” example, so not really a complete vehicle. And that was what I related to sibling vehicles. The idea was that, for example, when we have one platform, but two different vehicles, to be able to predict their behavior. So I had a toy FE model, and I was trying to see that, if I have a set of data – a development tree – that means that your simulations are related based on the changes, because you always do a small change.

For example, you pick one part and do a thickness change, you pick another part and add a hole in that. So [we] consider this tree, and then the physical changes relation, to see if we could transfer these behaviors to another sibling vehicle. With that, it means that the design changes are the same, it's just the impact mass is different. That means we have more kinetic energy in that one. Since we are in a high nonlinearity with crash deformations – when you add a lot more mass in that, you increase the deformations, and it really gets harder and harder when the difference of these masses is huge. But on a smaller level, we got quite interesting results.

But still, since we had a limited number of simulations, because I wanted to have it connected to reality – maybe you have a maximum of 300 simulations that you want to transfer. So what I was doing was quite interesting – it was pair learning. I was trying to use the connection of simulations and use the relation of the edges between the relations to predict the level of absorption in them.

32:03

Alexey

**Do I understand correctly that graphs not only give us a way of expressing and storing knowledge, but also, when it comes to actually making these simulations, we can also express… each node could be a different experiment and the connections between them on the edge would be the change that we make, right? Because if you make a small change, these two experiments are quite related and you can express that in a graph database that these are very related – there was only one small change between these two experiments and you can also record the outcome for this change, which would be quite difficult in a tabular database, I guess. [Anahita agrees]**

32:53

Anahita

Yes. Yes.

32:57

Alexey

**I imagine if I, again, talk about tabular format, it would be just changing one of the features and then seeing what happens at the end?**

33:08

Anahita

I think in this scenario, it could also be that – because we were looking at simulations – compared to all of the simulations, so you can still think that you kind of have it tabular in that framework. But I think what was quite interesting is that when the simulations have an edge, real connection, we know they are closer than the one… So it's a kind of additional information in having the development tree included.

# Combining Knowledge Graphs and Large Language Models (LLMs)

33:43

Alexey

**You also worked on the LLMs, right? You see how this knowledge graphs and LLMs can work together. So can you tell us more about that?**

33:56

Anahita

I think it's quite a niche topic now, with LLMs and knowledge graphs. That's really focused on text data. With my PhD, I really didn't focus much on text data (NLP). So that's quite a new journey for me. I started from Summer, after the bootcamp, where I did some more of the LLMs. Within LLMs and knowledge graphs, a lot of focus is on grounding the answer. So it's really in this part of hallucination of LLMs and to see how you could feed in indexes that will produce more reliable answers.

Also, I think that it’s quite a fun domain because I think there a lot of people are against knowledge graphs and LLMs, but I feel that graph data science is something that could also support it, because with this selected of ‘what is the most similar node to feed in from a knowledge graph to LLMs?’ It could also be relations that come from graph data science and edge predictions and so on.

35:16

Alexey

**Well, to be honest, I still didn't really understand the connection. [chuckles] So, LLMs work on text data and graphs work on graph data – on nodes and edges, right? And does that mean that… Okay, we have different features or characteristics of our experiments (of cars or whatever) which are text-based and we can use LLMs for these features? Or what exactly does the connection look like?**

35:47

Anahita

Yeah, one funny example was that like the questions that LLMs can answer. One of them, I think was like, “Person A gives birth to a child in Canada… and then the father is there…” What was it? I don't remember the example really well. But the thing is, what I wanted to [say]… Now I should come up with my own example, because I don’t remember. It will be a bit fun. But the thing is that, within the relations you have within… Because when we do LLMs, we have the part of text, we do a chunks in them and then, based on these chunks, you don't have all the data connections to each other.

Also, since LLMs are not good at having huge amounts of input together, then we would miss some of these relations of data to each other. The thing is, when we build a knowledge graph based on the text data, we are transferring (we are generating) semantics and adding those relations in the knowledge graph. Based on that, you could fit into LLM, so it's a bit more of prompt engineering in that direction – what you should feed in to get good answers from LLMs.

So within knowledge graphs, you feed in some more relations. For example, with this example that I was trying to say, you don't know about who is the mother of a child, for example, but when you have the relations of the people in a graph, and transfer that and get all the chunks of information about the mother and the father, and the child itself, you enrich the data to answer the questions.

37:38

Alexey

**Do I understand it correctly that we have a piece of text that expresses different relationships between entities and what we use an LLM for, is actually extracting these relationships from the text. And then, once these relationships are extracted, we can put them in the graph and then use a graph database or graph library to further analyze the relationship extract. No?**

38:10

Anahita

This is just used as LLM with vector databases. So you make chunks, and for each chunk, you use LLM to do embedding (to extract features) and then you store them in the vector database. [Alexey agrees] This is like you have the chunk base – so you don't have any knowledge graph on that. But one of the basic ones is that, for example, to build a knowledge graph on that, is that, for example, you are getting a book and instead of having just these 400 tokens (for example) storing the node, you also have “What are the chapters of the book?” So you store the chapters as extra nodes and then you say, for example, “Chapter two owns these chunks.” You also add the relation that “These chapters come after the other chapter.” When you are just having chunks in a vector database, you are missing the relations and also you are missing the semantics. So these two are coming – these extra relations and semantics are coming with knowledge graphs.

39:22

Alexey

**Now I think I understood what you said. You mentioned prompt engineering. When we come up with a prompt, and we want an LLM to answer a question about something, we also want to include some semantic information about that. Like in your example about the book, you can say that “This paragraph comes from this page. This page comes from this chapter. This chapter comes from this section.” Right? [Anahita agrees] It’s just a sort of extra semantic information that will help an LLM to give a better answer, right?**

39:56

Anahita

Yeah. And, for example, when you do a template for your prompt, you can define the relations that are important. You can, for example, put it like a cipher query as an example for one of the use cases and since it has access to the whole of the knowledge graph, it could make similar cipher queries to find information about its own use case.

40:23

Alexey

**There is a question from Roasting Chestnut. That's a very interesting nickname. So the Roasting Chestnut is asking, “It sounds like transfer learning. Is that the same idea used in relationships?”**

40:42

Anahita

In which content is he asking? Is it about…?[chuckles] Because the only transfer learning I have been doing is about predictions on automotive. I am not sure if that's the question.

40:52

Alexey

**I guess it’s like… You mentioned embeddings – making embeddings from using LLM, right? So we use the knowledge that is already in LLMs to make embeddings. Maybe this is it?**

41:05

Anahita

I don't think so. Because in transfer learning, a lot of the time you have some layers. Because with this LLM content, it's more of ROG, I would say – retrieval augmented generations. You use the embedding – you use a model that exists, and you will try to do the embedding on them. And within transfer learning… I think if I look at it from the window of deep learning, it’s always that you have a model and architecture, and you try to transfer it from one dataset to another dataset. That's in more of a level of fine-tuning, I would say.

41:51

Alexey

**The reason I thought about the example I mentioned – when there is a piece of text. An example could be that this is the mother… or different relationship between things in text, and we want to ask an LLM to extract this graph. Because recently, I was doing something a little bit similar. I had a piece of text and I needed to create a JSON from this text, so I needed to give it some structure. And then, of course, instead of doing it myself, I just asked an LLM to do this. And ChatGPT did it pretty well. That's why I thought, maybe we can actually also use it for not only extracting JSON from that, but also extracting relationships.**

42:42

Anahita

Yeah, it's quite… I have tested it a bit. Also, it's like with LangChain, I would say. It's kind of a tree of thoughts, where you ask a question, it gives you some items, and you go further down. The only tricky thing is that you need setups to work. Because if there is a limited amount of questions and depths to work, I would say it's easy to check it. But when you want to extract a lot of content from your text, it’s quite tricky to trust it.

A lot of times, with knowledge graphs that come with LLMs, it's exactly the opposite. You want to have something that you can trust. That's why I think, still, the old process of building knowledge graphs is required, because if you have it with an LLM, you're in a loop. You can never validate it. Maybe it can help you, but still, to verify it is quite tricky. With ChatGPT, you can also see that if you ask it for an output today, it could have different results after a while or with a bit different content – the history of it is affecting the answer that it's providing.

# Practical Applications and Projects

44:13

Alexey

**Right now, I'm looking at one of your GitHub projects. This project is called ADPTLRNFHYCSS…[chuckles] I guess it's “adaptive learning physics”. Right?**

44:31

Anahita

[chuckles] Yes, yeah. Quite a tricky name. [chuckles]

44:36

Alexey

**[laughs] So, can you tell us more about this project?**

44:38

Anahita

Okay. Actually it's interesting because it's quite connected to what we just discussed, because this was an LLM bootcamp project. It was the first time that we tried (for myself, as well) to connect knowledge graphs with LLMs. That was the idea of thinking that, nowadays, you can ask lots of questions from GPT. How it could be to have a platform or a piece of code that it can make as learning material. So our first target was to have it in physics. But, as it is with a lot of projects, you aim for something, and you end up on something else. That was exactly [what happened] as we had just two weeks to finish this project. A lot of hallucination and untrustworthy [results] from GPT on generating the content. How we were doing it, it was saying that…

Okay, so we had LangChain break down a topic, and we wanted to make a knowledge graph for physics where you can provide material to users. The first users come, and then you ask some questions about the interest, the levels, and so on. Then it provides you, step-by-step – content, question, answer that supports you to learn a new domain. That was the initial idea. But since it was building the knowledge graph, [it was] really hard, and really not good results out of ChatGPT (3.5 Turbo [chuckles]). That's when we decided to change the project a bit. Instead of having a learning platform for physics, we moved it to (still in a learning direction, but) supporting people to read papers.

If you are not a researcher, or if you're a researcher, how could we have a platform where you give it a paper, and it breaks it down to you, and it could support you with the same consequence, where you feel comfortable to read papers in domains you have never read. That was breaking down the papers to the sections, finding the relation between them, and also connecting it to other papers and the references, and so on. So that was the final result of it.

47:10

Alexey

**I'm looking right now at the visualization. What happens there is that there is this part where you upload the file? [Anahita agrees] That is actually all you need in this example. You upload this, and then, what it's doing is extracting text from the PDF, and looks at different terms there (different words), right?[Anahita agrees] And it builds a graph from these words and explains each of them, right? Or how does that work?**

47:44

Anahita

We have two tracks. In the first track, it's doing semantics. So it extracts each section of the paper, and then does embeddings on them, and finds the relations between them. And the other track is that it gets each section and uses some steps with GPT to find what the keywords for these sections are, and then defining the keywords. So it's independent of the graph. So it's with several prompts to extract more knowledge.

Then, I think at the end of this, you see that when you hover over some keywords, it should show you a small box where it's describing that. So it's more about having it all in one place, so that you don't need to move around to read a paper. Also, with these sections and these graphs, you click on the edges of them, and you can explode it and see what is common between two sections and what the differences are. This is more interesting when you have two different papers.

For example, you want to compare their method development – when you click on the edge, it explodes and summarizes the differences and similarities of those. This works only on archive papers, because really extracting semantics with PDF and getting the sections is not so simple. It could really differ how people are publishing and generating their PDF and with the archive it’s good that they always get the texts and generate it themselves, so it's more trustworthy to extract the PDF from them. Then what we do is, when you upload the paper, we could also get all the references out, and then make a summary of the paper and its references, and show what is, for example, the most relevant reference to that.

On visualization, we are using PageRank that these sizes of the nodes are referring to how important that node is in the whole network that you see. And then we could see that “Okay, what is the most important [thing] and reference within this domain?” This is a bit more help for people who are used to reading papers, because for them, it's quite time-saving to find the most relevant related work to dive into.

50:20

Alexey

**I wish I had a tool like that when I was doing my Master’s. [Anahita chuckles] I was doing information retrieval on mathematics – on mathematical formulas. And just reading… Especially at the beginning – doing the state-of-the-art research, understanding what is there and reading just random papers that are potentially about the topic, and then feeling completely lost there. I think I would have been more effective with a tool like that and I wish I had something like that. Also [inaudible]**

51:06

Anahita

I didn't hear you, sorry. [chuckles] I just didn't hear you. So I just asked “What did you say?”

51:14

Alexey

**Everything I was saying you didn't hear?**

51:17

Anahita

No, just the last part. Because I saw that you were talking.

51:20

Alexey

**I also wanted to mention that… I was taking a German test last week. And I was preparing for this German test. There are so many grammar concepts that are related – sample connectors, right? So it can be connectors that connect sentences, then there could be… I don't know how much German you know – some connectors there for dependent sentences. Sometimes these two sentences are independent and sometimes the connectors are actually adverbs, not real connectors. There are different sorts of connectors. And there are different sorts of grammatical things.**

**And for me, when I try to comprehend all that, my mind just blows [up]. [Anahita chuckles] And having this graph structure where… some sort of mind map? Where you can zoom in on different concepts and see how they're connected. That could be quite useful. I think with a little bit of tweaking, if I understand correctly, this project can also do that, right?**

52:36

Anahita

Yes, I think… I mean, it's a lot of time with this finding relation, if it relies on an LLM, then it [depends on] how well it's described and so on. But I think chunking grammatic pieces into a node and then finding the relations between them – it could be a great start. For me, it's the same all the time – learning new things, I need to find the relations between them. And that helps, like “Okay, this is the same as this one.” And when you have no clue and just walk around, it's hard to summarize everything and learn it. [Alexey agrees] That's the main reason, I think, we came here.

But the first goal of the project was, I think, even more interesting in this direction – where you can have a chain of questions going down that really makes teaching material, I would say. The reason we were calling it “adaptive” is [because it was] based on what your pace of learning is, these paths will differ. Because I think people have different techniques or different preferences. In that direction, instead of having a one-for-all teaching process, the platform can learn from you while providing more content for you. But the scope was too big for two weeks, I would say. [chuckles] Maybe later on, we could work more on that.

# Challenges and Learnings

54:18

Alexey

**What was the most difficult part in this project?**

54:23

Anahita

I think that [the biggest challenge was that] we couldn't automate generating the graph. That was the biggest challenge. Then, also, changing the project topic to something that is really tangible. So it's not domain-specific – that it will be interesting for people with no background or research and so on. Because that was one of the demands – to find a problem that matters for a lot of people.

54:57

Alexey

**For example, if I think about something that is not related to research and has a lot of different complex relationships (relations) between things, I think of Game of Thrones. Have you watched it?**

55:10

Anahita

[chuckles] Yes.

55:12

Alexey

**[chuckles] Like all these people who do different things, they belong to different houses and whatnot. When you watch this, you think, “What's happening there?” So having a graph there that explains what's going on would really be helpful.**

55:32

Anahita

[chuckles] Yeah, or which people are in which chapters and like things like that. [chuckles]

55:36

Alexey

**Yeah. Okay. Was the deployment part also difficult? I see that you used Fly.io. Or it was something that… [cross-talk]**

55:48

Anahita

In Streamlit it… We had fun – on the demo day, Streamlit didn’t work. So we deployed it on Streamlit and exactly at the time we started to show our demo live, Streamlit crashed. And I would say it's hard to develop with Streamlit when you go to more complex visualizations. So I think if we go back…

56:11

Alexey

**Like graphs, right?**

56:13

Anahita

With graphs, it was still okay, but when you have a sequence of dynamics – so you click here… So it's mostly state management and frontend development. Because when you have a lot of dynamics and interconnections, then it's getting quite time-consuming to solve this state within Streamlit. Still, it's great that it's supported. So you can still do quite complicated visualizations [chuckles] but it will take time.

56:47

Alexey

**But yeah, it's… The interface is still pretty advanced from what I see. There is a piece of text and then you hover over a term and then it explains what this term is. And I also see this graph structure, and then you can explore it, which is pretty impressive for a two-week project. Super impressive.**

57:11

Anahita

The thing is, we were two – so I wasn't alone. We both had experience in web development. Also, the other team member, Jonathan, had great web development experience. So, together, it was a lot to do [chuckles] I would say. Even though we failed on the first topic, so we've lost some time.

57:38

Alexey

**But it's part of the process, right?**

57:40

Anahita

Yeah.

# Anahita’s Recommendations

57:46

Alexey

**If… Do you know any books or other resources for people to learn more about graph machine learning, knowledge graphs, and how LLMs can be used for that? I don't know if there are many books about LLMs yet. Maybe not. But in general, maybe there are some cool resources that you can recommend?**

58:08

Anahita

I never learned with books. [chuckles] So that's kind of a tricky thing to recommend, really. But I could recommend courses. I really enjoyed the courses from Stanford for graph data science. Also, recently, Deep Learning AI had a really cool short course for knowledge graph and LLMs. I really find it inspiring. Within LLMs, to be honest, I think it is still… I mean to learn the basics, I think I like the course of Sebastian. It's not yet complete, I think… I really liked the content he's contributing. I also don't remember his last name. Sorry. My memory is never super-duper. I'm just trying to find the last name. But maybe I could give their full name later on. I think that looks quite interesting – the material.

Because the repo is not completed for LLMs either. More than that – it's always digging in and finding new content. Tricky to have it as an old-fashioned domain like mechanical engineering, where you read a book, and it's all in [there]. Because it's part of learning in this domain – to always have an ear out to look for new things, and also learning how to navigate with all of this information and not to feel overwhelmed while still moving forward.

60:03

Alexey

**And the course from Stanford that you mentioned is called Machine Learning With Graphs, right? [Anahita agrees] By Jure Leskovec. [Anahita agrees] He's quite a well-known person in the graph community, right?**

60:18

Anahita

Yeah. And I really also recommend following the Graph Conference. He's taking care of it. I think it has so far been a free conference to attend. The research exchange there is really great.

60:41

Alexey

**Yeah. We should be wrapping up. That's all we have time for today. So thanks a lot, Anahita, for joining us today, for sharing your experience with us – answering all the questions. And thanks, everyone, for joining us today, too – being active, asking questions. Yeah, it was fun.**

61:01

Anahita

Thanks a lot. It was fun. [chuckles]

61:04

Alexey

**Yeah. See you around!**