The Messy Middle of Large Language Models

featuring Jay Palat and Rachel Dzombak

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Rachel Dzombak: Hi, everyone, and welcome to the SEI Podcast Series. My name is Dr. Rachel Dzombak, and I am a senior advisor to the head of the SEI's AI [Artificial Intelligence] division. Today, I am so excited to welcome my coworker and friend, Jay Palat, a senior engineer and the SEI's interim technical director focused on AI for mission. Today, we are going to discuss large language models (LLMs) and what advancements for AI and large language models mean for software development.

Welcome, Jay.

Jay Palat: Thank you. It is good to be here.

Rachel: Jay, why don't we start off by having you tell our audience a little bit about yourself and the work that you do here at the SEI?

Jay: Sure. I have been in the industry for over 20 years now. I started when the web was young, around 1999, and I have seen a lot of different

technology curves come and go. So I have been here through web and Mobile Web 2.0., and now exploring the world of AI. My work today is showing how we can use and leverage AI for using in real-world capabilities. So, how do we apply AI to mission-critical problems, how to help people successfully use AI, and move it from the academic world into the government world?

Rachel: Well, certainly, seeing tech trends come and go fits right in with what we are going to be talking about today, which is the trends that we're seeing and the recent growth and applications leveraging large language models. When talking about large language models, most people think of tools that leverage large language models. Of course, in the recent months, ChatGPT and Copilot have dominated headlines. Despite all of those headlines, I always think some level setting is helpful as a starting place. For the benefit of our audience, can we have you just spend a little bit of time defining what you mean when you talk about large language models?

Jay: Sure. Large language models work on the way of trying to predict the next word. If I said to you, *See, Jane*, you couldn't naturally fill it in with a number of different words to finish that sentence. That is because we have a large facility for language, we understand things. In order to simulate something like that, large language models take large corpuses of information— think about large chunks of writing that's available on the internet—and learns patterns from it. What distinguishes large language models from earlier types of models is the way that they analyze and make these connections. In previous models, it was usually only making relationships between one or two words away. In our case, it would be, What is the relationship between *see, Jane*, and the final word, right? So it looked maybe one or two words back to build a relationship, which is great for finishing small sentences, but as the sentence got longer and longer, the models tend to hallucinate a little bit more and babble into two different kinds of nonsensical patterns. The thing that changed for large language models is being able to look at larger pieces of context, not just at the final sentence, but like all the sentences that came before it and came after it. If you think about it, folks may have played the game of Mad Libs growing up, right? Mad Libs is a game where you provide nouns and verbs and adjectives, with no context and plug them into sentences to create funny stories. It is essentially the same thing that large language models are doing. They are taking these different prompts; they are looking at these different words and finding connections between them. What makes Mad Libs funny is you can't see the context around it, so the words just kind of randomly come in. For large language models, they see the prompts, and they try and fill in different

words, and then they are scoring or trying to predict what is the best word that fits in there. Over time, using lots of examples, they learn to put sensical words into these Mad Libs-type structures in order to successfully build out longer thoughts.

Rachel: I love this notion of how the key shift was having more context to be able to determine that next sensical word. Could you go a little bit deeper into that? What enables that larger context? How are the models of today able to access that larger context? What are they drawing on?

Jay: The big transformations for all modern machine learning (ML) is having more and more data to work with. The transformation in computer vision was having ImageNet with like 14 million labeled images. For large language models, it has been being able to grab large corpuses of data. There have been a lot of books that have been published in human history, which have been turned into electronic formats that we can now consume. There have been more active models. So things like Wikipedia, things like Reddit. So there are corpuses of large amounts of conversations, which leads to some interesting behaviors. The large language models are learning how we speak and learning from the content on the Internet. Sometimes that information is accurate and correct. Sometimes it is not. People occasionally tell falsehoods, more than occasionally, tell falsehoods on the Internet. Large language models can't really tell the difference between them. So there is a lot of work that needs to be done to process information and turn it from just random information to more structured, curated knowledge that can be input into these systems. In the programming space, for instance, there was a lot of effort that went into taking information that came out of GitHub. So there were a lot of open-source projects that were released. People started building datasets out of that open-source information, which gave it a large corpus of semi-curated information of like these are projects with a certain number of stars or a certain number of followers. They used that as a criteria of, *Is this good information to be included in our dataset*? For some of the large language models, Open AI created a team based out of Kenya to do curation. So they had to go through the samples of language to figure out, were these appropriate or inappropriate models to work off of? But it takes a lot of time and effort to build these models to curate the data that makes them possible.

Rachel: Absolutely. And even at the end of that curation, we still see bias exists within the corpuses of data because the internet is not necessarily representative of everyone in all cultures, all populations, et cetera. Let's talk a little bit about how…OK we have some basis of understanding for large

language models. Previously, I mentioned some tools that are leveraging large language models, Copilot, ChatGPT. How do the tools that I mentioned leverage large language models, and what are common misconceptions that people have about those tools?

Jay: What has been interesting about the evolution of these tools is kind of what affordances are being created by them. ChatGPT was based off of GPT 3.5. It took some changes. So there was human-based reinforcement learning, which helped get more curation in an automated sense. What I mean by that is, what ChatGPT does is when you give it a prompt and it gives you a response, there is that little thumbs up, thumbs down signature that lets the application know that it was a good answer or a bad answer. Now, what it means by a good answer and bad answer can vary by context. So it could be it was an ill-formed sentence, it doesn't make any sense. That is a pretty simple one where, OK that is a model that we can look at and see if there are adjustments there. But it can also be, this information is incorrect. So there have been plenty of examples out there where ChatGPT is asked different questions and, with complete confidence, gives very incorrect answers. So knowing how to use that tool, part of it is understanding the material that you are asking about to see if it is correctly interpreting or putting together that information in the context of your question or prompts.

Rachel: Let's talk a little bit more about, I love that you said it is really thinking about how to use that tool. What are examples of applications that we should be thinking about ways we can leverage a tool like ChatGPT? What are places where we should not think about leveraging a tool like ChatGPT? This speaks, I think, a bit to the misconceptions piece of clarifying the context of use, et cetera.

Jay: There is a lot to unpack in that one. There are ways that people are excited using it today and trying to figure out where the right places or the wrong places are still, we are figuring out the balance of that. In the example, you gave of Copilot, there is an example, there of using it to generate code. The large language models have learned from lots and lots of GitHub repository open-source projects and are able to help predict like what is the next section of code that you're looking for? Sometimes that goes from things like *I want to use a documentation string. This function will do X, Y, and Z*. The model will be able to generate an appropriate function that will do that. That was an affordance that took time. The original format that they started with was something along the lines of ChatGPT, where you would give it a prompt and it would give something back. What they found was that the creators or some of the sponsors of ChatGPT or, sorry, Copilot in earlier

iterations had done something where it was a question-and-answer format, and they found that they had what they considered spooky or kooky. Sometimes the answer came back so great it was spooky. They were just amazed at how good the quality of the response was. And sometimes the response came back super kooky of like no human would ever make this mistake. Even someone who was like a basic programmer would know not to do whatever they had tried to do or what the system generated as a response. Over time, they realized that that caused a lot of distrust of the system. It doesn't matter how many good responses you get. If you get some really bad ones, you start distrusting the system and moving away from it. So they changed the format so that did more of an autocomplete, where it would fill in the text in continuation. It is an area where the affordance makes a difference in how it becomes acceptable to use the tool. There is a correctness in programming. You actually have one program that will run and execute the algorithms that you are looking for. Some of the ways that we treat this is like we can use tools for checking the correctness of these different programs. There are lots of tools that we use with new human programmers, code reviews, or static analysis that helps us build better trust around these systems. But there are a lot of areas where we haven't built tools for establishing trust. That is kind of where it is better to be more cautious on using these systems.

Rachel: I know that one of the things you and I talk about a lot is interpreting information and how people interpret information. We have talked about the mix or the perception that ChatGPT is a search engine and always predicts the right answer versus an augmenting tool, something that can help with exploration or give additional information. So, for me, I think about using tools like this as a starting place. The other night, I wanted to think about a recipe. So I asked ChatGPT to give me a recipe that leveraged beans, eggs, and sausage because that is what I had in my kitchen at that time. It gave me that one recipe that then I iterated with a little bit. That is really in the space of AI augmenting human behavior versus replacing it. So I realized that probably falls into your kooky category of provocations. But could you react to that a little bit? How do you think that behavior shows up in workplaces today as people are starting to think about using tools such as Copilot or ChatGPT?

Jay: I think it is going to create a lot of places where people need to think about how do they fashion their work. In New York City, there was a prohibition on using ChatGPT in schools. They wanted to make sure that students are learning to think to write essays properly, and so they are trying to prohibit the use of these tools in order to make sure that students aren't

using them to cheat their way around the system. Some educators have taken the opposite tack of using ChatGPT to create a baseline of, *Here is an essay that it is created, correct it. Find the errors, either factual, logical or argument errors that the machine has made and build better arguments around it, create a better essay based on this template*. We are still figuring out what are the right ways to do that. There was an article about a physician who has found that the best way you can use ChatGPT was not on diagnosis, but when a physician sees a patient and diagnosis a problem, and then recommends a remedy for them, right, a prescription, that is not the end of the story. Sometimes in giving a treatment, the physician then needs to talk to an insurance company in order to make sure that the company is willing to sign off and approve that remedy. To do that requires time of the physician to write these long letters that describe the need, the treatment, and why it is the best solution forward. What this physician found was that instead of spending 20 minutes writing these letters, he put the information in ChatGPT and it would write a reasonable letter that he could send to an insurance company turning like a 15-minute writing assignment into like a 5-minute assignment where you could spend the difference in time with the patients rather than trying to work with the insurance companies. There is a lot of places where people have figured out how to add it to their workflow to enhance and augment rather than trying to replace the human in the loop.

Rachel: I love that example of the physician because I think the real opportunity here is to find alignment between problem spaces and tools such as this as a solution. Right now, people are exploring everything, and saying, *Oh, how could we use this to fundamentally replace core behaviors such as diagnosis*? But actually, there are tons of applications that are perhaps less shiny at the outset that have a huge potential for change over time. On that note, you talked a little bit about how schools in New York were fearful and saying, *Oh, this is going to replace our kid's ability to think*. Parts of that are valid in many ways. You could see skills plateauing because of overuse of tools. But when you introduced yourself, you said that you have been through several waves of technology trends. I am curious if you have seen this reaction before and what it reminds you of, of past technology developments. Or, is this current wave of focus on large language models fundamentally new and different?

Jay: I don't think it is new and different. It is a new technology and shiny, and there is a lot of excitement around it. I think that is true. I think the adoption of these things has been faster as people are getting used to riding these curves. But in some ways, it is very similar to previous technology adoptions. The first phase of any technology adoption is, *How do I do what I'm doing*

today using this new tool? When the web first came out, there was a lot of graphic designers who were trying to design websites like magazines. It had to be specific alignments. It had to be specific ways of using fonts, specific placements, and it had to look like a static image. And it didn't take advantage of the richness of the web. So when we look at what was known as web 2.0, what was happening was the real creation of native applications for the web that took advantage of that responsiveness and used it in new and different ways. One of the first items that came out was Google Maps. We have had maps before. We have had maps for hundreds of years. But having maps where you could zoom in and out and be able to place things and be able to move around them and then get to a finer level of detail with just a little bit of scrolling was something very novel and different. It added a richness of information that allowed for new applications to be built on top of it. There became then this thing of, *How do I use location data in new and novel ways?* People were starting to figure out, *How do I integrate maps in my applications, not just for location?* but then going further with things like applications where we could check in with your location. There was a popular set of applications that were all about being spotted in different places and dropping notes about what you are doing there. The first generation of any technology is like, *How do I ape the one that came before or the ones that I know well*? and then becomes the next level of, *How do I use this in new and novel ways that take advantage of the strengths of the platform or the media rather than trying to ape what came previously?*

Rachel: I love that. I think it is really thinking about it as a palette and a tool in the toolbox to drive change versus saying, *This is the be-all, end-all, we are going to focus on it*. It is what new behaviors can unlock because at the end of the day, people are solving problems instead of technology solving problems. But towards that end of thinking about what's new, what's different, of course integrated development environments (IDEs) have had code generation and automation tools for years. What do new advancements in AI and large language models mean for software development?

Jay: For me, I think what is most interesting about it is how this opens up for exploration of new language or new technologies. Today if you want to learn a new language, you go out and you check out some blog posts or maybe a book or tutorials on YouTube and watch how people are developing code in a particular language. Say I wanted to learn Rust as a language. It is new. It is popular, there are a lot of blog posts and books out there. I could pick one up and start working through it. But Rust is kind of well-known for its difficulty of memory management. It takes a little bit of new ways of thinking to use it well, and so getting to idiomatic Rust takes a little bit of time. What is

different about these tools is they can provide ways of letting you experiment and get correction in a more intuitive fashion. If you make a mistake, some of these tools can explain what a line of code is doing. So you can look at examples and not just read the code for what it is but get annotations to it. Having a ChatGPT explain line by line, what a program is doing, might help you understand the code base faster, might help you understand the language faster. So I think it gives like better affordances to understanding ways to tackle problems and new languages or new technologies.

Rachel: Absolutely. You spoke a little bit earlier about trust and the role that trust plays in the adoption of these technologies. Certainly, in our work on AI engineering with the Department of Defense, we think a lot about calibrated trust and what assurances are needed to feel confident using AI and high-stakes environments. If ChatGPT recommends me the wrong recipe, no harm, no foul, I'll just move on with my life. But when thinking about trust in large language models today, what considerations come to mind? What are the types of things that you are thinking about?

Jay: I mean, ChatGPT can still make mistakes, and if the recipe goes horribly wrong, the consequences can still be grave. As we think about building these tools and using them, there is trying to figure out how do we calibrate trust and what kind of scenarios we use them in. In programming, I think it is important to recognize that while ChatGPT has a wide vocabulary of languages and a lot of good examples to work from, it can still do the wrong thing or that understanding of the author's intent may not reach where it needs to be. So we can use tools that we use today for evaluating, helping new programmers or people on our team to code better to use these tools appropriately. So things like code review, it is not enough. You can't just roll out and say, *OK, ChatGPT gave me this answer. Let me plug into my new application*. It still requires the understanding of the code so that you apply it appropriately, and it is using the context and you understand, is it actually meeting your mission need? Are you actually achieving the goal you are going for? Does it have the same concerns you do? There is an interesting thing in experiments I've done and others have done with ChatGPT of like where it makes subtle issues or errors. In a blog post recently posted, we gave an example of having ChatGPT write a haiku. From first appearances, it looked like a good haiku, but in reality, it was not using the patterns that haiku traditionally use of the 5-7-5 syllable pattern. It recognized it was supposed to write short sentences, but it used like a 7-9-7, which is not the format for haiku, even though it looked like it could be correct. There are a lot of those types of errors where they are subtle, and you really have to

understand what you are expecting from it and know how to check your answers before you should apply these to mission-critical problems.

Rachel: What you are getting at there, I think is a level of critical thinking around the responses. I also think about the ways we are doing critical thinking around trends. Certainly right now, family members are asking me what I think about the rise of large language models as well as some of the organizational leaders that we work with. I am curious, as you are tracking recent developments, what sort of trends are you seeing that give you a lot of excitement or make you feel optimistic about use in applications? And what are trends that give you pause? What are places where you say, *Oh, I think people should be paying a bit more attention to those subtle failures that existed in that space?*

Jay: On the excitement part, I got to admit that is a little bit easier sometimes to think about is…

Rachel: Absolutely.

Jay: …looking at the ways that people are taking these models and applying them in different directions. What I mean by that is, there are folks who are looking at large language models as like, *How do we create these long sequences of interactions and apply them in new and different ways?* Some of them isn't traditionally what we think of as language. There is some interesting work in genetics where people are doing protein folding, biology, where people are looking at, *Can we use these large language models to simulate protein foldings in novel ways, so that we can get to faster discovery of new drugs or new proteins to help us solve large-scale illnesses that we haven't been able to deal with so far?* There are folks who are trying to do things like communicate with other types of languages. Moving beyond just human language, but also the language of animals, looking at those patterns to figure out, is there ways of predicting what is coming next to do translations in places that we have never done translations before. There are folks trying to use it to recover lost languages of things that are being done like in archaeology, where we have like patterns of language that maybe we can discover new things about. Being able to translate things where we don't have Rosetta Stones to discover language we haven't touched before. Lots of places where there are these interactions of patterns where language and conversation can also be worked in as these back and forths.

Where it is not as comfortable or where I think that people need to be critical is still these are not perfect knowledge bases of information. They are

gathered, many of them, from sources on the internet, and sometimes the internet is wrong. To your point about critical thinking, depending on the type of knowledge you are looking for, you need to be conversant with it and understand what is coming in in order to understand what's coming out. By itself, it is not just a search engine. A lot of the ChatGPT models have been built in isolation and don't have access to new information. So you get these really strange bottled results that are as true as the information was when it was released in 2021, which is where some of these corpuses end. So if the world has changed since 2021, and odds are that it has in different areas, the models may not be accurate anymore. Being able to understand the currency of the information, being able to understand like the patterns. In the Stratechery podcast, the host talked about the first time he used it, he was looking at a history argument of, I believe it was for John Locke, and where he stood in politics. The answer was completely wrong. It was looking at things that were said about his work, but it took the exact wrong opposite tack to what it should have been. There are a lot of places where you really need to know your material to master it and understand that it really doesn't understand context. Some of the work that people are doing to try and improve results involves changing the prompts for them. Sometimes it is amazing that the quality of the answer can be changed by the prompt that you are using and giving to these machines, which is not something we are used to in our systems. So far today, it has always been, *I put in like a search query and I get the same results out*. Minor variations don't make that much of a difference. But with the prompts, there is a little bit more of you can get very strange results or very more accurate results, depending on how you ask the question.

Rachel: So, again, it is a reflection on behaviors, of finding new ways to interrogate, play with, to explore, and to think about how large language models and the tools that are built on top of them can shift our workflows for better or for worse, where they are applicable and where they are not. So say you are listening to this podcast, you have heard about the hype around large language models and you want to start exploring further, what resources would you recommend? Where would you suggest someone starts if they are trying to learn more about large language models?

Jay: How to use or how to make them?

Rachel: Great question. Well, let's think about how to use them first.

Jay: Honestly, now is a great time to experiment. There are a lot of things out there that people are doing and captivating and sharing. I found it interesting

the other day that Anthropic AI, which is a company, has posted a job posting for a prompt librarian. Their job is to look and curate the different prompts and the results that come out of it. They are not sure if this is a permanent job, but it is a piece of what they're looking at for their future. They are integrating in so many places now. I mean, ChatGPT is available as kind of like the OG source of everyone's exposure to large language models. But even Microsoft is getting into the act of integrating it with Edge, the Edge browser and their Bing search engine. I think a lot of those programs are in beta, so you may need to wait to join. But there are a lot of communities coming up now of people discussing prompts, talking about the work they are going through. I think the interesting one of the side spaces is the generative art. So looking at DALL-E and Midjourney, there are a lot of communities looking at how do we create new types of digital art based on prompts. What is interesting about this is there is a lot of community and sharing about what makes good prompt and good prompt engineering. Definitely in the graphic design space, I think people are still trying to figure out, *How do I build good large language models for ChatGPT*? But they are starting to build communities now of people talking about what makes for good prompts. Almost as interesting is how do you fool these systems, like how can I create counterintuitive prompts that get out of the safety measures that these tools have been built with?

Rachel: Absolutely, and let's switch to the other side. If I am interested in building on top of large language models, where would I start on that front?

Jay: There are a lot of different avenues for entering that direction too. The open-source community has been building some tools and libraries that allow you to talk to ChatGPT through their web interface. Open AI has started releasing new API models that allow people to communicate with the different models, specifically with the GPT model that underlies ChatGPT. If you want to use it locally, there are transformer models that are available. Meta recently released their LLaMA large language model, which is something that can be run on a single GPU. So it is accessible to more people than just like researchers who have large cloud accounts. Hugging Face is a repository for different models, and they have a number of transformer models available for people who want to experiment and try writing their own applications on top of a large language model.

Rachel: Lastly, Jay, is there anything that I didn't ask you about that is important for our listeners to know about large language models?

Jay: I think it is a really exciting time to be using them. As people explore and

talk about it, you need to have that curiosity. There is an overwhelming amount of hype right now. People think it can do anything. It is the ultimate oracle that is going to give you the perfect answer to everything. And they are not. Large language models are built by people on top of human-curated knowledge. There is nothing superhuman about them. They have, to your point earlier, all the biases that come from the people that are building them today. There is a lot of unexpected behavior coming out of these models because there is a lot of unexpected behavior that comes out of people. There is no universal set of, *This is truth, this is bias*. You know, I had a student asked me the other day about, can't we just mark everything as like a bias bit and say, *All right, this is going to be where it is doing something wrong*. There isn't universal agreement about what is a bias, what is a good thing or a bad thing. There are different values that are being encoded in the system in ways that we don't really understand yet. So that is part of the learning and growth that we need to do as both creators of these systems and users as understanding that they are flawed and figuring out how that we work with those flaws and understand them better. To your point, getting to that calibrated trust.

Rachel: Jay, thank you so much for being here to talk about large language models. To our listeners, thanks for joining us today. We will include links in our transcript to all the resources that we mentioned during the podcast. Finally, a reminder to our audience that our podcasts are available on SoundCloud, Stitcher, Apple Podcasts, Google Podcasts, as well as the SEI's YouTube channel. If you like what you see and hear today, give us a thumbs up. Thanks again for joining us.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, and Apple Podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please do not hesitate to email us at info@sei.cmu.edu. Thank you.