ChatGPT and the Evolution of Large Language Models: 4 Case Studies

featuring Matthew Walsh and Dominic Ross as Interviewed by Dominic Ross

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.

Tom Scanlon: Hi. Welcome to the SEI Podcast Series. My name is Tom Scanlon. I am a technical manager for the CERT Data Science team here at the SEI. I am joined today by Matthew Walsh, a senior data scientist on my team, and Dominic Ross, a multimedia design team lead here at the SEI. We are here today to talk about large language models (LLMs) and some use cases for them. Welcome, Matt and Dominic. I can't wait to hear what you have to say about this. We are going to do some introductions. I'd like to start with Matthew. Matt, you are new to the SEI podcast series. Can you tell us a little bit about what you do here at the SEI, how you ended up here, and your interest in large language models?

Matthew Walsh: Sure. Yes, my name is Matt Walsh. My background is in computational neuroscience, so creating computer programs that simulate how the brain changes as people learn. In the past, I have applied that to training and education to create technologies to enhance learner experiences. The current wave of AI systems make use of deep neural

networks. That approach is loosely based on the architecture of the human brain. It is exciting for me to think about the human brain as a model to create more intelligent computer systems. At the SEI, do work in that area and applying those technologies to cybersecurity.

Tom: Thank you. Very interesting. Dom, you have done the SEI Podcast Series before. Welcome back. Previously, you did a talk on deep fakes, which we'll put a link to that in the transcript for today. That is another type of generative AI. We will probably touch on that today too. Can you give us the same introduction? Tell us about yourself, what you do here, and how you ended up here.

Dominic Ross: Yes, of course. I started at the SEI roughly 15 years ago, specializing in training and education and experimental classroom instruction. From there, I diverged from classroom training to more film-style production, and with that came generative AI and the components that underlie how you can actually achieve higher-quality results with lower budgets. What we did was we first moved into deep fake technology, and from there, that was my dipping my toe into generative AI. After that, I started moving into more text-to-image prompts, and, from there, text-to-text prompts. It is a fascinating field where there are a lot of underlying themes that overlap. Happy to have my kind of foothold in the next form of my digital career here at the SEI.

Tom: Great. To get going already. Already we have introduced a couple of terms to the audience: generative AI and large language models (LLM). Could you tell us a little bit about what generative AI is, what large language models are, and then sort of comment on why they are in vogue now and why people are so interested in them. Language model has been around for a long time. I think I learned about them 20-plus years ago. Now they are really emergent, so if you could comment on that too.

Matthew: Yes, so let me begin by giving a 10,000-foot view of this space, and then I'll ask Dom to comment on some of the particulars of large language models. Most generally, artificial intelligence involves creating computers that can behave in rational ways, in ways that used to require human intelligence. One of the ways to imbue a computer with the knowledge that it needs to behave in a rational way is by giving it demonstrations and examples, by training it to acquire the rules to perform a task. This is called machine learning. So machine learning is a subset of artificial intelligence. In the past several years, the big trend in machine learning has been the use of deep neural networks. This is a particular approach to training machines that

is more flexible and scalable than methods that have been used in the past. Deep neural networks form a subspace of machine learning. Generative AI is a type of machine learning or an application of machine learning. The real breakthrough there came with recognizing how we treat the output of a given task. Traditionally, with machine learning, outputs were things like given a set of pixels and a radiological image, is there a medical diagnosis? Given different factors about a house, what is its likely sales price? In the case of generative AI, researchers realized that they could relax that definition of what an output is. Given a sentence like *See Spot…*, the goal could be to predict the next word in that sentence and *run* would be the next word. In the case of generative AI, rather than predicting a single outcome, you predict something like the next item that will appear in a pattern, and you can apply that over and over again to generate new sentences, new passages of text, and entirely new readings that didn't previously exist. This is an example of using machine learning to generate something new. That same approach can be generalized to things like creating images that didn't previously exist, creating audio signals, and creating video signals. So machine learning has gone beyond now just predicting an outcome to generating entirely new content.

Tom: Thanks, Matt. That is fascinating, and I think a key point is with generative AI, we are talking about generating content. We are not just using AI machine learning to make a prediction, but we are generating novel content, images, paragraphs of text, any type of content.

Dominic: What we have seen now is the value in being able to generate these predictive text models. But also, the trade-offs that go with it. There is the ability for some form of hallucination. Where a model, because of the way that it predicts the text, sounds very assured in its response. An average individual sometimes looks at that response and trusts it at face value. When you are dealing with large language models, at least at the moment, one thing to keep in mind is if you go to Wikipedia and source a topic, you may be not want to trust the response that you get back initially. You have to do your own research. We are getting into human out-of-the-loop and human-in-the-loop problems. There is no ability for this large language model to transform to an autonomous task at this point. But someday, there might be. We really need that human in the loop that is observing these responses and correcting them as they go along.

Tom: I heard you both say, when you describe the language model, that it is predicting the next word or the next words in the text. What makes something a large language model? How does that differentiate from the

language models we have been using for a long time?

Matthew: I am going to step way back in answering that question. Begin with the fact that language is a distinctively human ability. From a developmental psychology perspective. Language emerges almost universally. All people in all parts of the world naturally acquire language. From a comparative psychology perspective, no other species' linguistic structure approaches the complexity of human language. When thinking about language from a psychology perspective, this is a distinctively human ability. In terms of AI research, you can go back to Alan Turing's 1950s paper where he proposed a task to have a human interacting with a computer or another human trying to judge whether it was artificial intelligence or human intelligence that they were interacting with. When he proposed that task, the mode of communication was through natural language. Even from the earliest days in artificial intelligence, there was an emphasis on natural language processing. In the 70 years since Turing proposed that test, artificial intelligence research on language has undergone numerous paradigm shifts beginning with symbolic language processing with handcrafted rules to statistical language processing where the rules were learned much as humans learn rules, to the current wave of NLP applications, which use deep learning. As compared to earlier statistical approaches, the deep learning approach is more expressive and scalable than statistical methods that existed in the past. This is the road map that has brought us to the current set of language models. Dominic, do you want to talk a little bit more about questions about how large is large for an LLM, and what makes them more distinct from the historic approaches that have been pursued?

Dominic: Sure. The foundational models that you are alluding to are typically the ones by Open AI, Google's Bard. There is a large open-source community that is exploring models specifically with Llama. What most people have been introduced to is a model that is somewhere between 7 billion to 65 billion tokens. That is essentially how much language is actually included in the model.

Tom: When you say a token, what is a token in this context?

Dominic: A token is simply a word. A token is a value that gets returned as part of the predictive model. What we see, though, is not necessarily the larger the model with the higher quality results. Whenever you are evaluating large language models, you have to be careful about what type of data it was trained on. Most of these models scrape the internet for their data, or they had some type of proprietary data that they uploaded when it comes to fine-

tuning. We will get more into fine-tuning later in this broadcast. But one thing I want to talk about is size versus what an average person can use at home. Most people experience this through cloud computing. Cloud computing, with a lot of these big tech companies, has infinite resources. They have access to thousands of GPUs [graphics processing units], these huge arrays, and they can load these very large, large language models onto them where you would need 48 to 96 gigs of ERAM. What we find, though, is the larger the model, the slower the token response. Why is that critical? Well, if I am researching a topic, and the predictive text is coming through as 1-to-2 words per second, as opposed to 15 to 30 words per second, I am going to get quickly distracted as well as frustrated with using this model. There is a little bit of an art in the algorithm, as we talked about in the previous podcast that I was in, where you have to understand what your use case is and how you want to deploy this large language model. Then, from there, access it through parallelization or over separate arrays, so that you can sparse out that data for quicker responses.

Tom: Matt and Dom and a couple other researchers, you recently put out a whitepaper on some use cases and then a corresponding blog post. Those will also be linked. We are going to talk about a few use cases in a few minutes. Before we get into specific use cases, though, with any new exciting technology like LLMs, there are challenges, and there are opportunities. Can you talk about the opportunities, in general, that people should be excited about using LLMs? And then what are some of the challenges that we are facing right now with LLMs?

Matthew: [To Dominic] Do you want to take a shot at opportunities, and I will speak to some of the challenges.

Dominic: Yes, sure. The first opportunity is really around hyper creation. So summarizing content, analyzing data, researching, vast topics. It is a really great source for administrative-style tasks. Then we get into the hyper-creation that you alluded to before where we are maybe creating multiple design layouts or writing code.

Tom: Can you explain hyper-creation as a term. So how is that different than creating content? How is that being modified?

Dominic: It is the rapid pace of creating content. If I am a designer, and you ask me for a design mockup, I may be able to create you five unique concepts. In that same amount of time, I could have a large language model create maybe 100 different types of concepts. Maybe you go with concept 56

that I never thought of. That is hyper-creation because now, at this point, as a human inside the loop, I can actually go through and refine that idea and tailor it so that it is a much more rounded representation of what you were originally looking for. When I am talking about hyper-creation, that is what I am talking about. The same thing that comes with transitioning art. There is Midjourney, for example, which is a generative of AI tool, that has the ability to take text and create totally unique graphics near instantly. From there, you can actually find the parameters to remove, to filter, to generate different types of looks, themes, artwork. That is where hyper-creation really comes in. But it also comes with code. There are a lot of developers that are now starting to use some of these, I call them plug-in tools, that tie directly to a large language model's API. From there, they are able to start writing basic syntax. That I start writing a comment and from my comment, it interprets what I'm looking for and post a response. And there, as the operator, I am able to choose whether or not I want to accept that response.

Tom: To summarize what I think I heard you say is working at scale, creating novel or new content, and also sort of brainstorming or helping us develop, in human-machine teaming, create new content.

Dominic: That is right.

Tom: That sounds very exciting, and I can't wait to use something like that. Matthew, what are the challenges we face?

Matthew: I want to begin with this concept of artificial general intelligence or AGI. The theoretical idea of a system that can perform as well or better than humans across all tasks. We take the example of ChatGPT or other LLMs, and they are not AGI. However, they do have great breadth of capability across tasks, and they are very good at performing some tasks. As humans, it is easy to misattribute the LLM's abilities and overestimate its abilities in different areas. There is another concept of anthropomorphism, and this is imputing human characteristics on an object. The way that ChatGPT is designed, when you interact with it, it does have human characteristics, both in its responses, and in the manner and tone of the way that it interacts with the user. So it is easy to begin to impute human characteristics on ChatGPT. Again, it is not, in fact, human intelligence. This can cause us to have missed expectations about ChatGPT's ability. For example, it can describe all manner of classic chess players and advances in chess, but it might not be able to describe the rules of chess. Whereas if we are talking with a human, we would expect for them to have that base knowledge if they have very specific knowledge about a given topic. I bring these two things up, AGI and anthropomorphism,

because they lead to classic examples of misuse. This is a human overestimating the abilities of ChatGPT and becoming overly reliant on it when they should be more critical of its outputs. I want to give three examples of misuse that involve GPT's knowledge, and this is not just true of GPT but LLMs more generally.

The first concerns gaps in its knowledge. All of these models are trained up to a given date, and they don't necessarily have access to information that was released beyond that date. This can come up in the case of academic publications that are very recent and code libraries that were recently released. We as humans may know something about those, but the model may not have included that in its training set.

The second are factual errors. When you are interacting with an LLM, you might have the expectation that, in effect, it stored away all of the things that it was exposed to, as in a database, and it goes and accesses those pieces of factual knowledge. In fact, they are represented in a pattern of weights that the model has acquired. Those weights don't just depend on a particular piece of scientific literature that it reviewed, but everything else that it was exposed to. As a result, those weights can drift, and at times, the model's factual knowledge can be inaccurate.

Then the third source of knowledge error is confabulation. Really the LLM GPT is trained to complete sentences. If you ask it a question and that question draws on something that it wasn't trained on, it will still complete the sentence, but it will complete the sentence in a way that is not based on any existing knowledge. This is kind of the origin of confabulation. One of the things is because GPT gives such convincing answers due to the way that it was tuned for human users, there is no indication in its response that it is confabulating.

These are three sources of knowledge error that we can run into with an LLM. That is on the topic of misuse. There is also the concern for abuse. This is using a system intentionally for malicious purposes. Just as we can use LLMs to do things like generate content at scale, an adversary can use an LLM to generate misinformation and disinformation. They can take a particular political figure and do so in a way that very closely mimics the tone and the mannerisms of that individual speaking. Further, you can stack that with other forms of generative AI, like synthetic audio and synthetic visual, in order to create a transcript and then a video of a political figure saying something that in fact they didn't. This is an example of abuse. Other examples of abuse are prompt injection. So the user submits a prompt to

GPT or some other LLM, and if an adversary has access to that prompt, they can include additional context that will cause GPT to return an incorrect answer. Then the final form of abuse that I want to talk about is called a jailbreak. These models are trained on potentially sensitive information, and there are ways to abuse the knowledge contained within the model. For example, asking GPT to create a script that has malicious intent when deployed on a computer system. Now there are guardrails put in place to try to minimize the occurrence of that happening with something like GPT. But a jailbreak involves crafting a prompt that causes the LLM to go outside of those guardrails and to get it to give you malicious information.

Tom: Dominic, what else would you add about potential risks associated with LLMs?

Dominic: Well, I would like just to build on your jailbreak to begin with. What has become very popular recently is prompt hacking competitions where there are specific scenarios that are developed, and how can I manipulate a prompt inject to get a result that I want? A good example was I was at a recent conference, and they had created a conference prize for actually having an LLM tell you a piece of data that their guardrails were specifically set up to protect against. It is surprisingly unsophisticated how these models are manipulated. I don't know if I want to go into the different techniques here. What I would really like to say is adversaries are always going to find workarounds to foundations put in place, as well as bias can be created because of those guard rails or foundations put in place. That touched on a number of topics where data sets may not be trusted or misinformation may be created. But also bias can be simply created by under representation. There is a large contingent of biases that have happened in the generative AI space from the imagery perspective based off the data sets that they were trained on. When it comes the large language models, there are also underserved communities that are actually also being negatively impacted. Not seeing representation in the responses that they are given, which I would say is a huge, huge risk to moving forward.

Tom: Thank you both. That is a very thought-provoking on some of these challenges. One of the challenges I want to pull a thread on a little bit that you both alluded to. Matt, you mentioned that the models are trained on data as of a certain time. You both talked about the guardrails. The models that most folks encounter like ChatGPT and Bard or pretrained models, but there is code out there for folks to develop their own large language models. I want you to talk a little bit about sort of the process of creating a large language model beyond just training the model. There is a lot that happens

to the publicly available models like ChatGPT and Bard in terms of putting those guards in and fine-tuning the interface, the prompt interface. Can you talk about just a little bit about that challenge of making a model usable?

Matthew: Yes, let me take a shot at describing the tech stack and the end-to-end training pipeline. We have used this term *foundation model* several times during this podcast, and I just want to comment. The foundation model is not just in the area of language. There are visual foundation models. There are graphic foundation models.

Two key points about the foundation model. First, it is trained on a tremendous amount of data that gives it great breadth. Second, specific applications are built atop foundation models. Those two characteristics evoke this image of a physical foundation. The language foundation models by themselves don't naturally lend themselves to use. There are additional steps that need to be taken to tailor the language model for a particular application. The first of those, if you want to take a general language model and deploy it in a specific domain like medicine, you have to take a body of content knowledge specific to that domain and then fine-tune the model. It begins as kind of a general learner and you turn it into an expert in a given domain. It is still the case that once you have done that fine-tuning, if you ask the language model for responses, it won't naturally give you responses to the question. Keep in mind, it is trained to predict the next word or sequence of words in a sentence, not to answer a question. There are additional steps that involve instructional tuning in order to nudge the language model to the point that it responds in a useful way. What does this look like? It might look like creating a data set of prompts with examples of the types of responses that you want to receive. The model is then trained on that data set, so that it can respond to prompts in a way that is beneficial to the user. It might involve reinforcement learning. Here the language model actually produces responses to prompts, and humans essentially give the responses thumbs up or thumbs down. This is, again, to nudge the model in the direction of responding in a way that the human expects and needs to perform their task.

Tom: If I am a user of one of these models, and I know that you are doing all this fine-tuning as a model creator, how do I get trust in the results that come back in? How do I validate? As you mentioned, the model is not originally intended to answer questions. It is just to predict text, right? But we are using it to answer questions. When I get an answer back, how do I determine if I should trust that answer?

Dominic: Well, Matt already touched on the reinforcement learning, and that is a key component, but also, a lot of these models have the ability to present you with cited sourcing. One thing you have to do is just be a good investigator and don't trust anything at face value but go and actually validate the responses that have been provided to you. If you think about these foundation models, four out of the top five are U.S.-based and built by big tech companies. We have limited access into what those data sets are. We get a little bit more insight into Meta's open Llama, where we have the ability to work with that from a more fine-tuning perspective. You see an open-source community actually generate, I think, tens if not hundreds of fine-tune models off this Meta open-source platform. The community, itself, is the one actually going through and doing the validation, responses, the evaluation. Checking going through and determining whether or not the response is giving a particular prompt inject or the tailored responses that you want back. There are no real assurances currently. The best thing that we can do is keep using this technology, evaluating and improving, and fine-tuning it as we go.

Matthew: Yes, and if I could build on that and echo some of Dominic's points. When we go to use the model, how can we have any confidence in its outputs? Some of the things that Dominic said and some additional things that come to mind…The first is to use bespoke training data. In the case of code generation LLMs are trained on open-source code like code contained in GitHub. One of the problems with that is code contained in GitHub may have errors, and it may have security vulnerabilities. To be more confident in the output of an LLM used for code generation, you can train it on a curated set of code. To be more confident in the output of an LLM in the context of medicine, you can train it on medical texts rather than training on Wikipedia. So you can curate the data set that you use to train the model to be more confident in its outputs.

I think the second is task selection. In the case of code generation, you can have automated tasks in order to validate that the output produced by the LLM, the code output, is functionally correct. Then, you can run it through security tools to detect vulnerabilities. There are certain use cases where you can validate the output of the LLM. An adjacent point is the manner of use. If we are using an LLM for scientific research, you can ask it questions about the papers that were included in the training set. Alternatively, you can direct it to a PDF of the paper and make use of the LLM's linguistic processing ability to generate a summary directly from the paper rather than asking it to, essentially, retrieve that summary from its pre-trained weights. The matter of use is one other way that you can control the quality of its output.

Then, the last thing is to ask the question multiple times and look at the variation in the LLMs output. If there is wide variation in its output, then it is a less reliable output.

Tom: Okay. That is great insight. Thank you. There are definitely some concerns and challenges with LLMs. We just highlighted them. I want to shift focus back to the opportunities. We are getting asked a lot here at the SEI about how can people use and leverage and leverage LLMs to work better, faster smarter. You mentioned some use cases in the conversation like code generation. specifically, in the whitepaper that you published, you outlined four use cases, opportunities to use LLMs. Could you talk a little bit about how those use cases were developed, and which use cases you came up with and what opportunity you see for those use cases?

Matthew: Yes, sure. To tie this back to your question about the risks, we talked about the potential for misuse. We talked about the potential for abuse, but a third risk is the potential for disuse. There are times when it would be beneficial to use this technology. If we wait until the risks are driven down to zero, then we will disuse the technology and miss out on those opportunities. In the white paper, we developed a set of four use cases. The first use case was focused on code generation. Within the space of code generation, providing skeleton scripts and having GPT complete the scripts, presenting GPT with error messages and having it debug code, and then giving GPT completed code and having it generate documentation to go along with the code. One of the things that I will note there is in the case of completing code, GPT will sometimes create syntactical errors. We can't use it unintended on autopilot. We have to have a human involved reviewing the outputs of GPT to ensure the quality of the code that it produces. Nevertheless, it still allowed us to arrive a functional code in less time than if a human is working alone.

Tom: There is probably a trade-off between using an LLM to create more code. You might have to increase your testing and validation a little bit, but you still may see a net gain in productivity.

Matthew: Exactly. Yes. The second use case focused on education, and here we used the LLM in multiple levels of the hierarchy. The use case focused on creating a curriculum for a data science class. At the highest level of the hierarchy, we used the LLM to generate the set of topics that the course could be comprised of. Then we drilled down a level and had the LLM fill in the contents within each of those modules. Then, we drilled down one level further and had the LLM create the actual question bank and score student

responses. This is a way to accelerate curriculum development. But in terms of developing and administering a curriculum, some of the most costly things are developing the assets, so the content that is delivered in the course. Then, a second thing that is very costly and time-consuming is providing personalized feedback to students. Within the tightest cycle of the learning loop, we see an opportunity for the use of LLMs to generate spoke content for individuals and to provide personalized feedback to individuals. Aside from creating the overall curriculum, the LLM could enable personalization in a way that is not currently possible. Dominic, do you have other thoughts about the use of LLMs and generative AI in training and education?

Dominic: Well, specifically, when it comes to upskilling the current workforce, there is a large demand to fill multiple work roles that are not necessarily available given a myriad of factors. LLMs are really great at leveling the playing field. Where if you can take somebody that has strong soft skills, say in communication, and really lend themselves to being advanced. When it comes to prompt engineering, you can take those skills and translate them directly into getting tangible benefits from introducing a large language model into the workflow or even into their educational path, so that they can learn those hard skills as they go along while keeping pace with the current workforce. Data analysis is a huge one. There are just not enough machine learning scientists in the world. Being able to filter out that data, publish it in a way that is readable and digestible by all that are involved, is something that an LLM can really help [with] when it comes to, again, upscaling that current workforce. You can also talk about providing that personalized feedback. See, *I am designing a list of questions*. *I have this conference coming up.* I can actually provide that list of questions to a large language model and ask*, Are these acceptable? What am I missing? How can I improve this content?* Whether I am writing a resume or a blog post, there are a number of aspects to these large language models that can be beneficial. I think they say 60 percent of workers, actually, their productivity goes significantly up with the introduction of these models. Where we have to be careful though is releasing proprietary data. Even with training and education, but also when it comes to implementation into a workforce type of application, we have to make sure that we are not sharing information with these models that we don't want to get out in the world.

Tom: In the training education, you used the term bespoke a couple of times. That might be a niche term for some folks. That just means custom or fit for purpose, in our context, personalized. You talk about personalized feedback to learners, but the LLM could also enable, potentially, personalized training in the first place, right?

A lot of people get individual training plans, and we talk about LLMs being of being able to help us work at scale. You can scale up custom training and education.

Matthew: In the educational psychology literature, one of the classic findings is called the 2-sigma problem. This is the finding that when people receive one-on-one tutoring from another human, they do two standard deviations better than their peers. But it is very costly to deliver that one-on-one tutoring. So personalization and personalization enabled by LLMs, may be one way to deliver that level of individualization at scale.

Tom: In a cost-effective manner.

Matthew: Exactly.

Dominic: Yes, well, it definitely lowers the barrier of entry. As an individual, if I am going to my direct manager, I may be self-conscious about asking certain types of questions that I may be willing to share with a machine that doesn't necessarily ever get tired of answering the same response over and over again.

Tom: We have hit on two great use cases already, code generation and the training and education. What are the other use cases?

Matthew: The third use case focused on research. When embarking on scientific course of study or when developing a new product, an important prerequisite is to perform a landscape analysis. What other work has been done that is related to this topic, and how can we build on that work? The third use case focused on research using an LLM to perform that lit review. We went about that in two ways. The first was to query the LLM based on data that was included in its training set. And to ask it things like, *Historically, what are some of the approaches that have been taken to ensure the safe and responsible use of AI and machine learning*? This hit on querying the pretrained knowledge contained within the LLM. The second approach was to make use of the linguistic abilities inherent in the LLM and to give it new papers and new reports and to ask it to do things like generate a summary of this paper; generate a list of the main themes in this paper; perform different types of thematic coding on this paper. Again, it goes back to the model was pretrained on something, but it may be inaccurate when it retrieves that information versus relying on its linguistic abilities to have it generate summaries. The benefit of having it generate summaries of a given PDF is

that you can also ask it to cite its evidence. If it is relying on information that is stored in its pre-trained weights, it can't give direct citations to support its responses. Whereas when it is applying its linguistic abilities to a paper, it can provide that direct support for its assertions.

The fourth use case was focused on strategic foresight, and this is looking at some new and emerging technology and then thinking about how it could be useful in the next 5-to-10 years. The first three use cases involved procedural knowledge and declarative knowledge. But this fourth use case requires more creativity than the previous use cases. One of the drawbacks that we mentioned with LLMs was the tendency for confabulation. That is making an inferential leap that goes beyond what was included in the data that it was trained on. In many cases, that is a liability, but when it comes to creative tasks, confabulation can actually be an asset. In the case of strategic foresight, we are asking the LLM to take information that it has been exposed on and to extrapolate that to high-leverage use cases in the next 5-to-10 years. This is particularly interesting because given any unusual query, for example, *What are some emerging technologies in materials science*? *How can those technologies be leveraged for training and education*? It will respond in creative ways. Many of the ideas that it comes up with can immediately be dismissed, but some of the ideas are evocative. This use case on strategic foresight highlights the creativity of LLMs. Another benefit of using LLMs in this case is overcoming social conformity. When performing strategic foresight, people may be resistant to object to ideas that are being proposed. But here you can use the LLM to overcome that social conformity bias and encourage new ways of thinking outside the box and challenging ideas.

Tom: I also see an opportunity to use custom messaging for different folks that may object the same way we talked about customized learning.

Matthew: Yes.

Dominic: Well, yes. Another use case would simply be translation services. So you are moving into customized messaging, but also the ability to create the content in a language that feels natural for the learners themselves.

I think what Matt has really touched on with all these different use cases is the microcosm of where we are starting to see LLMs being financial, health care, institutional positions, software engineering. It is refreshing to see that given a strategic and academic approach, you can actually see how these can be implemented without causing an individual to have to worry about losing their position. More time allows for better responses, which allows for more

productivity from the particular individual. I think Matt has really touched on a lot of applications that lend them to the future of LLM implementation.

Tom: Great comment, Dom. In your paper, you and Matt highlighted four use cases, and then here in the conversation, you have thrown a few more on the table. One point I want to make is these four use cases aren't the only four use cases. There are many, many use cases for LLMs. For our audience who may be getting excited here in this conversation and want to get involved with LLMs, we want to enable some transition and let them be able to use LLMs. How would they get started? How would you get into working with LLMs?

Dominic: Well, I would say the first thing would be, don't be afraid to explore the current solutions that are out there. There are a number of free offerings that individuals can dip their toe to test the water. There is Google Bard. There is Microsoft Bing where you can use a search base LLM and actually curate content that you are looking for on the web. Matt touched on real estate analysis. If I want to find a target neighborhood and a five-bedroom house with three bathrooms with a certain school within walking distance, I can ask an LLM for that type of application. We also have the ability to use some of the pay-for-service, LLMs. ChatGPT is great; 3.5, I believe, is still freely available. If you need a more sophisticated version, you can go to ChatGPT 4, but it is really not being afraid to explore them. That is what I really want to encourage. There is a lot of fear within the risks that LLMs propose, but the average individuals that we see adopting LLMs are the younger generation. Digital transformation is hard for everyone. We all struggle with implementing new technology in our day-to-day life. I saw a fascinating talk on prompt engineering given by a 16-year-old girl who was just as sophisticated as any data scientist that gave the same talk later in the conference. What we are seeing is the same introduction with LLMs that we saw with mobile computing technology. Traditional desktop computing moved to touchscreen tablet-based computing, and the thought process was*, Well, how long is it going to take the average individual to adopt this technology*? Where we found children, as young as two and three, were able to pick up the technology and master it within a very short amount of time. Using the technology is the best way to get introduced.

Tom: Using technology is a great way to get it. Any other thoughts, Matt, on what you might do? Maybe you are using it, and you have questions or you get stuck.

Matthew: Yes, so I agree that a good first step is to make use of publicly available LLMs. Once an organization commits to incorporating LLMs into

their workflow, a next step is to undertake workforce education. Doing things like educating the workforce about potential benefits of using LLMs as well as the risks and precautions that workers can take to mitigate those risks. The final step in the transition would be for the organization to take advantage of open-source LLMs and deploy one or more on premise. There are resources available like H2O. It is possible to access open-source LLMs, deploy them on local computing resources, and then make them available to the workforce. To summarize all of that, making use of publicly available LLMs, educating the workforce, and then eventually committing to standing up an LLM on-premise.

Dominic: To build on the continuing education piece, it is not necessarily the workforce, but it is also an organization that needs the continuing education opportunities. You have to educate the C suite. They have to understand the benefits of the LLMs, but also the infrastructure and cost and risk and benefits of a potential LLM. So they are willing to trickledown those use cases to their workforce environment.

Tom: That is a great point, Dominic. You want to make strategic use of this technology. You have to plan for it, budget for it. It's not going to happen by accident.

Tom: You have both done tremendous work in preparing this whitepaper and the blog to educate people. I am curious, Dom, what is next for you in this space? What are you going to work on next?

Dominic: I will be primarily transitioning into what Matt alluded to, which is the localization of open source LLM infrastructure, the tech stack, the LLM stack itself for institutional use, but also generative AI with use with virtual production and gamification of training and educational content.

Tom: Very interesting. I can't wait to see what insights you share in the future. Matt, what about you? What's next for you in this space?

Matthew: Yes, two areas of interest. Going back to your earlier question about how can we trust the outputs of LLMs? One topic that I am working on is developing tasks and test harnesses to provide more formal assurances about the LLM's behavior. A second area of interest is to leverage LLMs for code generation. As I mentioned before, LLMs are trained on open-source code. Sometimes that code may be functionally correct, but it may contain vulnerabilities. One of the topics of interest is developing methods to allow LLMs to generate code that contains fewer vulnerabilities.

Tom: OK. Very good. Thank you both for your time.

I want to remind the audience that all the resources we mentioned throughout the conversation today will be linked in the transcript for this podcast. Also, a reminder to our audience that all our podcasts are available on Soundcloud, Stitcher, Apple Podcasts, and Google Podcasts, as well as YouTube on SEI channel. If you would like to see more of this content, go to today's recordings and give us a thumbs up and tell us we did good, and thank you everyone for your time today.