# AI is Stimulating Change

The last two years have brought seismic changes to the entire computing eco system unlike any we’ve seen before. This tectonic shift has been caused by large-scale research and development efforts in the computing disciplines of artificial intelligence (AI) and machine learning (ML).

These technologies aren’t new. Amazon has used ML to promote items of interest to you while shopping, your mobile phone completes words as you type with decent accuracy, and customer service bots have been able to handle tasks like returns and order statuses without human intervention.

What’s new is the capability, flexibility, and accessibility of these tools to companies and developers that don’t have the size or resources of large companies like Apple, Amazon, or Microsoft, etc. Basically, any developer with a small budget and a few simple scripts can begin adding these to their application with minimal effort. This is where we hope CODE Magazine will be a resource for our readers. This issue begins to deliver on this hope.

First, Sahil shows you how to use the content of voice audio passed to an AI model called Whisper, which translates speech into text. The text generated by Whisper is then passed to a large language model (LLM), which answers the questions posed in the speech model.

After that, Wei-Meng Lee does a deep dive into a toolset called LangChain. His article explores how you can create powerful AI pipelines by processing chat requests and refining the results using different models, techniques, and technologies. Not only does the article establish a good foundation for building AI pipelines, but it spends time discussing optimization, memory management, and performance. This is THE article you’ll keep next to you when it comes to adding AI features to your application. I know I will.

These articles are joined by several others, including using a new Copilot+ computer, exploring Semantic Kernel, and a really fun one on building role-playing encounters. This issue has a little bit for everyone!

When it comes to AI, the general focus of these pages concentrates on showing you how to use these tools and techniques in your applications. We’ll also attempt to provide discussion of the issues surrounding the impact of AI on us as developers, users, and society in general. These technologies are controversial—to say the least—and we’ll weigh in where appropriate. We started this process some time back in the May/Jun 2023 editorial “The Power of the Pivot” (<https://codemag.com/Article/2305011/The-Power-of-the-Pivot>). In that editorial, I reviewed a keynote at QCON London where Leslie Miley discussed some of the pitfalls of AI when it comes to power consumption, land use, and equity in general. This talk was given nearly two years ago, and its lessons resonate to this day.

That leads me to a topic that’s at the top of my mind. Can we trust the answers given by AI enabled applications? My answer is: PARTIALLY. Or, as the sign says in the pawn shop window, “Buyer Beware.” My reason for this answer is that it’s been shown that these AI applications can generate incorrect and sometimes “hallucinated” applications. For instance, one of our writers entered his name into a chat application and it turns out he is the co-founder and editor of CODE Magazine. That was definitely news.

In the big picture of things, this is a trivial example. Some AI implementations can cause real issues. Take the article “What Air Canada Lost In ‘Remarkable’ Lying AI Chatbot Case” from Forbes Magazine (<https://www.forbes.com/sites/marisagarcia/2024/02/19/what-air-canada-lost-in-remarkable-lying-ai-chatbot-case/>). This article shows how Air Canada lost in small claims court when a bot delivered bad information to a customer. We’re seeing article after article with examples of misuse of AI. There are movie trailers that include made-up quotes from real critics, ads with misleading answers to election questions, and the list goes on. It doesn’t take long to find info where an AI application has hallucinated some information that seems to apply—on the surface.

Given all this, why did I give AI a trust level of PARTIALLY? It’s because for some uses, these AI models are just good enough, provided that some human reviews their output for accuracy. As an example, **Figure 1** shows some Python code I was working on in VS Code.

A black background with white text

Description automatically generated

**Figure 1:** Some Python code and what VS Code’s AI recommended

The grey part of the code is what was recommended by VS Code immediately after I hit the space bar after the last character I entered. Should I rely on this code without consideration and examination? Absolutely not! It’s up to me to validate the accuracy and suitability of the code I write. How did it do? Not too bad, in this humble developer’s opinion. Helping coders be more productive is where these AI applications shine. Programming has many well-known concepts and techniques, and these AI tools do a fairly good job of incorporating them into developer tools. When it comes to output from these AI tools, I follow the old Reagan concept of “Trust but Verify.”

What really matters in the usage of these applications is suitability for the job. As a developer, you need to determine whether usage of AI has non-trivial risk associated with its responses. This is where we will be facing challenges for some time to come. In the pages of CODE Magazine, we’ll try to raise these questions and discuss potential answers to them. Now it’s time to use AI to build an encounter for that skeletal dragon mini I’m painting!