The Problem With Artificial Intelligence Isn't the Science, It's the Application

Joe Procopio

Yeah, I'm gonna talk about War Games.

Artificial Intelligence gets a bad rap, and as someone who has spent the last decade applying AI, I totally understand why. The science of AI and machine learning is solid, defensible, and utterly useful. It's how we're applying that science that rarely makes any sense.

This isn't a new phenomenon and it didn't start with AI. We're in the stage of the evolution of artificial intelligence in which the technology has surpassed our formerly rock-bottom expectations. When this happens, as has happened with other emerging technologies in the past, we start leapfrogging those expectations to cause two problems:

1. We produce dystopian stories of future state. In this case: The machines destroy us or rule us.
2. We start looking to cash in by applying the new technology in the most haphazard ways possible.

Both of these outcomes are expected and ill-advised, so here's what we do about them.

Our New Machine Overlords

I don't want to talk too much about the dystopian future of AI, I just want to make sure we're all on the same page about what AI really is so we can best figure out how to use it. That starts

with making sure we all agree what it's not.

Computers don't think. Most of what you read about the machines rising up and deciding humanity must be wiped out is bunk. Everything a computer does is, ultimately, designed and programmed by a human. Self-consciousness is not an option, scientifically speaking. We don't even know enough about our own self-consciousness to put the machine equation together. We're guessing. This is why we create what seem like self-fulfilling prophecies of doom.

But keep in mind, this same stopgap measure — that machines must be programmed by humans — is also its own greatest weakness. This is what Elon Musk and his ilk are really worried about. We humans can surely design systems with enough automation and enough stupidity to produce some scary outcomes.

This is not a new concept. In fact, the now-ancient movie *War Games* nailed the stupid human scenario almost perfectly, when our military computer decided that the game against the Soviet military computer — the game that launches nuclear missiles and then the world blows up — needed to be finished. Ultimately our computer had to be taught that no one wins that game. With tic tac toe.

Oh. Spoiler alert. See it anyway. It still holds up.

But the *real* problem with that kind of doomsday scenario is much simpler (and less Hollywood blockbuster friendly). If we're going to design nuclear launches controlled by computers, let's make sure it takes two humans to LAUNCH the missiles, not require two humans to STOP the automated launch.

War Games did this. What would have blown up the world is that quick scene where one military launch key guy is pointing a gun at the other military launch key guy who was hesitating turning his launch key. If there's an issue with that movie, it's that that scene wasn't explored, because the next three seconds of what those two humans do decide whether or not the movie takes a huge left turn and vaporizes us all.

And THAT, in an exaggerated nutshell, is the problem with how we're applying AI today.

Well, again, it's two problems:

1. Artificial Intelligence, Machine Learning, Automation, Chatbots, Alexa, all of it, is best applied, and probably only usefully applied, when we automate only those parts that can be perfectly automated.
2. AI, ML, blah blah blah, is best SOLD as magic that can automate any task that any human can do, because humans are expensive and make mistakes.

Are we that far removed from "voicemail hell" that we've forgotten the pain of all-in automation?

We might be. I see those promises made all the time. You do too, in any Watson commercial.

Why Do We Replace Amy?

We attack our AI projects by trying to replace expensive, mistake-making humans, immediately and completely, with machines. I'll give you a great example: AI scheduling assistants. Great

idea, questionably executed.

Anyone who has tried to get two or more people in the same place at the same time has dealt with scheduling hell, having to go back and forth several times to nail down the exact time everyone can meet. The use case for automation here is totally valid and the solution is totally valuable. The science is also there, since most of our 9-to-5 is documented digitally and that calendar data is at least available, if not public.

It is super, super easy to write the script that can offer options, take requests, even prioritize some preferences, and slap a block of time on two digital calendars.

So why does this fail more often than it succeeds?

Because the expensive, mistake-making executive assistant wasn't the problem. That part of the system wasn't even expensive, nor mistake-making. In fact, I usually run into the same scheduling problems whether the executive assistant is a machine or a human.

Because the problem is actually the executive.

Well, again, it's several problems. But here are the top three:

1. The executive rarely blocks off all the time on their calendars that needs to be blocked off. Unless it's a dedicated meeting, it's likely not on his or her calendar. And it never will be.
2. There are too many moving parts around the executive. Schedules are documented but they change. Priorities are undocumented and they also change.
3. Outliers rarely get programmed into the system. I once tried to have a conference call with someone overseas, and his AI assistant did not understand that I wasn't interested in doing that call between 1:00 a.m. and 4:00 a.m. my time. Instead of giving me midnight or 5:00 a.m. as an option, it just kept pushing the same times farther out on the calendar.

Ten seconds to text my colleague, ten seconds for him to text me back. Problem solved. In fact, he was gracious enough to schedule for 6:00 a.m. my time.

It's the Human (Dummy)

The missing piece is what I usually refer to as logic engineering or decision design around a complex system. Neither of these concepts are new, but they take on a new relevance when applied to AI, tackling some of the same stuff that happened with web, mobile, etc:

1. AI (and so on) is a new science. The people with all the AI science experience have little, if any, application experience.
2. The people with the application and business experience might be either dismissive or fearful of the science, or they just don't have the time to understand it.

This is why teenage Matthew Broderick had to train the military war computer not to blow up the world while old-ass Professor Falken had to stand behind him, providing guidance and smiling smugly while the world was seconds away from blowing up.

OK, parts of the movie don't hold up.

Especially when you think about the fact that Professor Falken should have coded for that contingency in the first place (see the scheduling-bot outlier problem above) or at the very least NOT AUTOMATED that part of the process.

This wisdom gap is only one possible reason for skipping the logic and decision engineering stage, resulting in AI application failure. There are tons of others. Money is another one. FOMO is another one. Malfeasance. Ignorance. And on and on. Oddly enough, it's the same kind of tug of war happening around blockchain and Bitcoin, with just as many misunderstandings and myths, and two very distinct, very dug-in sides on either side of viability.

My point is, whether it's Global Thermonuclear War or grabbing coffee with a colleague, there's a new science here in decision and logic engineering, one that's playing catch-up in the AI world. It's around the application. It's not in whether we automate, but where and how we choose to automate.

Automation, AI, ML, data science, the best of it will always involve the 80/20 rule, and you can invoke that rule all over the place. Systems should be 80% automated, 20% human. The path to get there should start with the reverse and work its way through the most complex problems first. Subsystems should be analyzed as 80% automatable, 20% non. Viability rules should be 80% norm, 20% outlier.

Those numbers aren't exact and the lines of division are always moving, but that's the beginning of logic engineering. When we forget to approach AI solutions with those types of application breakdowns in mind, that's when we get dystopian (or stupid) outcomes.

So how do we stop the stupid outcomes? We develop sound AI implementation strategy that bridges the gap between the science and application. Read more in my next post: Understanding How To Implement an AI Strategy.