3 Interview with Ted Benson

[00:00:12] Sil The last year has given us many new buzzwords. This includes "Agent" the idea that if you give the language model access to external tools, it can answer complex questions more competently. To learn more and to find out whether "Agent" is a buzzword, I spoke to Ted Benson who has a Ph.D. in Computer Science from MIT and is founder and CEO of Steamship, a startup specializing in building and scaling AI agents. He spoke to me about agents, AI startups and how he envisions the tech industry moving forward with AI.

[00:00:47] Ted So if you think of the word agency, what it means to have agency is to have some authority to make decisions on your own, to do things on your own. And that is approximately what it means in computing. It's an old term in computing, it goes back to the eighties. It's not something new with respect to LLMs or AI today. And it really just means some autonomous piece of software which is making some kind of decision for itself.

[00:01:14] Ted So where you see this in the industry now can be a full spectrum of, Hey, AI do a million things and figure it out, and then you kind of twiddle your thumbs. Those make great demos, they're not really production quality yet, but it can also mean something like a support chatbot where the support chatbot might be making the decision to look up a record in a database and tell you the answer. It might be making the decision to give you a coupon. The important part is that a human doesn't necessarily need to be involved for all of it in the way that maybe with a lot of software a human would be today.

[00:01:49] Ted So they've been around for decades with respect to language models. Some of the experiments that got people using the term again were when folks started using it kind of like an iterative loop where the language model was asked to come up with a plan and then execute each step in the plan. A lot of folks use that to mean agents, and it kind of equated the two back in the early days of Web programing, a lot of people equated Ruby on Rails with Ruby is another example of this. And so, you know, maybe__ and ChatGPT are examples of this.

[00:02:15] Ted I think more generally some people are calling it the "agentic" model of programing. I'm a software engineer, so I think of it from a software engineering perspective. It's one where the LLM has some capacity to invoke tooling that the programmer has endowed it with. And so this is a form of embodying, the tooling might be to generate an image for you, the tooling might be to ____ the tooling might be to log in to Upwork and hire a contractor to assemble an army to take over the world. In all of these cases, it's some capability that the LLM is in endowed with. That the LLM can then self reason and say, Hmm, I think the next step for me ought to be this.

[00:02:56] Ted And that reasoning loop, I think to your question of when the people start really talking about these and where they're going, I think when people started building these reasoning loops, that's when it really started entering kind of the popular conversation again. Because you can give the LLM a high level objective and there's certain ways you can coerce it to have this reasoning loop and then break down each step into some invocation or multiple invocations of some externalized tool, it tends to be, get the results of that, think again, and then decide what to do.

[00:03:23] Ted And so the answer to does it work or does it not work? Which I think the real and the attached really important question is sort of yes, no, like it depends completely
on the complexity of the problem, how good the LLM is, how good the thing is. And so we're kind of in a Wild West right now where I think it's definitely true. I think it would be hard to argue with the fact that this is going to have a lot of mileage as an industry that we're going to get out of it. I think it's also true that we're not entirely sure what the contours of it are yet in a production capable sense, and it's changing fast. And so it's definitely kind of the bleeding edge of the line between demo wear and production.

[00:04:04] Ted Well. So, I mean, good, good means different things. And I think right now we've definitely gotten into the territory where folks are very comfortable with production LLMS for creative applications. You know, tell me a story, talk to me as if you were a pirate, these kinds of things. And that's actually big business right now. Think of it as a new form of video gaming.

[00:04:24] Ted The hallucination problem for fact based industries, which is a lot of us, medicine, reporting, and so on and so on, is still a big deal. And I think that nobody yet seems to have a one size fits all solution for it. Lots of people are pursuing a variety of different solutions for it, from retraining to kind of aligning the model towards some kind of belief system or some kind of constraints, to just using multiple models to fact check, to changing the user interface around how we consume the output of these models so that the user is more equipped to have an understanding of whether or not it's lying. But nobody yet, I think, has the answer with a capital A and it's something that we definitely are going to have a lot of work to deal with because in some ways the same thing that causes the LLM to hallucinate, is the thing that causes the LLM to be so powerful and adaptive in general.

[00:05:24] Ted One of the hardest things right now, for folks in the industry and not in the industry, is to get a good sense of what's possible and what's not. You know, if I can jump, I don't know, four feet on the ground, it's easy to imagine therefore I should be able to jump four feet, two inches. And we kind of have that natural sense of extrapolation for so many things in life. I think what works and what doesn't work in AI is really tricky.

[00:05:48] Ted So I interact with a lot of folks who want to build things, because I run Steamship, which is an agent hosting platform that people use to build things like newsroom bots and other things. And one of the common situations that we run into is folks coming up with, call it like a requirement list, like we need it to do X, Y, Z, that any one of those things might be an interesting conversation and have a lot of realism to it. But taken all together, it's a much bigger task than they realize.

[00:06:15] Ted So I think my biggest piece of advice is to start small. Like think about the full breadth of things you could do and then kind of pare down and go step by step. Because if you try to do it all at once, the technology is still immature enough that you're probably in for a rocky road, whereas if you try to do it one thing at a time, just in my experience from the streets so to speak, I think that that's probably the path towards success.