Transcription - Jamie Wadley GPT Part 2

Welcome back. I'm Kim Baillie, she's Fulyana Orsborn and this is Inside Exec.We're continuing our discussion this week with Jamie Wadley on all things GPT and if you haven't heard the first part I would encourage you to listen to that first of all because we talked about the basics, about understanding different versions, different types and what it is useful for. This week we're going to look at it a little bit more in depth, about how you get it to do things, and that's called prompting.

It's a phrase, or a sentence or a couple of paragraphs, apparently, that you put in to instruct the large language model, the LLM, to provide you with some sort of response. So we might ask Jamie first of all, to run through for us, what prompting is and to give us, perhaps, a couple of examples so that we can understand it better for those of you who haven't used it up to this point in time.

Prompting is how we interface with these large language models, and it's not dissimilar to putting in a search query into with a search engine, but I found with my prompting, I think like most people, they start off fairly simple, usually in the form of a question. They can be so much more than that. I think of a prompt, myself, as a cross between computer coding and poetry. By asking the right question you can generate a much more powerful answer.Some of the earlier prompts I played with in the beginning, I used to be amazed at the differences if I asked in a slightly different fashion. It might be in the form of asking it to give you an answer and giving it a list of things it needs to do to give you that answer.

As you were talking I was thinking that there might be concern about the word "question" because in my experience of using it, very limited experience I will say, I wouldn't ask it questions. I would ask it to "do" something for me or I will "instruct" it to do something for me. I might say to it, please write me a poem, 16 lines about a topic, rather than "can you write me" a poem about this particular topic. I would tend to think of it more as instruction rather than questions. That's probably semantics and he's smiling at me at this point in time and it's lucky Fulyana is not here because she would be rolling your eyes.

I guess it's just how you approach the model. I, probably incorrectly, personify it quite a bit and I'm in the habit of saying please and thank you when I ask it to do things and I think that's true, you do give instructions to do things and some of the things I've had the model do have been quite intricate instructions. But really what we're doing is we're asking it for an output of some kind, whether you call that a question or whether you call that instruction is, as Kim says, that just semantics. The term that people use now, there's a new job position has become a thing now that these models are becoming more widely used, is "prompt engineers". These prompt engineers are people who have a skill to write a question for these models to generate the most concise, the most rich output that it is capable of. After you've used a model like GPT for a week or two, I think you'll find that the complexity of the things you write will be vastly different than they would have been the day that you first started to use it.

Does it become then, more of a conversation?

I think that's true. It is very conversational but ideally with the right prompting, the

conversation is almost one sided because the words you put in can lead to an output up to a thousand words, probably not quite that many. For a complex problem that I would use GPT for, it's not out of the question for me to write between 200 and 500 words of instruction. I usually give it instructions, by way of first summarising what I need it to do and if it's a complicated task, I give it a list of things that I think it needs to do first. It might be first, research keywords for me that are significant to the topic, review those keywords and find semantically similar keywords that can be used in creating a piece of internet copy to use on my website. These things I'm talking about are more based around search engine optimisation - it's something that GPT does very well. You can get very granular in what you want it to do and for the most part, it will do it.

Kim mentioned a poem that is 16 lines long. It's funny, the models have trouble with getting things like that correct. If I say I want you to write on this that is 500 words long, chances are it wont even get close to that because the nature of the way the output is created is that it generates the next word and it stops when it's finished (generating words). It's very hard for it to know whether that last word is word 500 or word 328 or word 642 because it's not counting the words, it's simply generating the answer for you, one word after another. It doesn't know what that answer will be. The machines themselves, though they are very clever, they don't have the ability to think like we do insofar as we reflect as we think. They think in a more linear way where they start the process and it takes them from one point to another. So, while there might be some sort of planning involved in what it creates, it's not like we can come up with, a poem for instance, saying it's going to be this format and it's going to be this many lines and this is the result. It's going to write one word after another. I heard it described, whether this is relevant or not, it's about like when we recognise someone's face, we don't look at their face and reflect on it. It's a linear process where we look at the face, we see someone, we recognise them, we know who they are - that's a linear form of thinking. The LLMs do very much the same thing insofar as they don't think about what they're thinking, but you can, by prompting, help them do that. Once again, this makes them more powerful but that's probably a story for another time.

With the prompting, those of you who have been long time listeners will recollect when we first spoke with Jamie many years ago, that he talked about a background in computing and in coding, in writing code. If we look at that for your particular instance, does that make it easier for you to think in the prompting fashion?

I don't think it helps me. It doesn't hekp me write the prompts themselves but maybe it gives me a bit of an inkling on how, with this linear process that I just mentioned, will roll out. Because in programming, back in my day when I was programming, essentially what you are writing is a list of instructions that you need a program to perform. And for the most part it followed a simple process of these are the inputs,tThis is the processing, this is the output. I probably think of the prompts of written in the same way but I don't sit and write code that becomes a prompt. I might well write a list of things I want it to do but it's far more about being accurate and maybe a little bit colourful for what you're telling it to do. Rather than say, I need you to summarize this text for me, you might ask you to, please make a concise summary of this text. And what we're doing is to push it a little bit harder to create better output for us. It makes a big difference, it's funny you wouldn't think it would. But certainly by making the prompts better, you will get better results. Give it a week or two you will find your prompts are vastly different from those you used on day 1.

Beautiful segue because, as you would know if you've listened to us for any period of time, we're very big on accurate speaking in these podcasts. Thank you for mentioning that you need to be accurate but I do want to explore that further because accurate speaking is something that we, as a species, are lazy about. We will use the "all, every, never" phrases at the drop of a hat to explain away istances in our lives that are not working the way we want them to. In this new age of using computing power that is at our fingertips, one way or another, accurate speaking to my mind is going to become more important. If we rely on GPT for a lot of our work, our basic work, we need to be accurate about what we're using as the prompts and that then should flow on to our interpersonal relationships and speaking in terms of our workplaces. With a prompt, if you are not accurate at the first prompt and we talked about it being conversational, can you refine it as you go along?

I think I've found that in that conversational mode, you do refine the answer as you go.

Do you get a refined response? So you ask a basic question, you use a basic prompt, we'll use the poem as the example - write me a poem about the garden. It writes you a poem and then you decide that it's not quite right, so you say write me a poem about the garden in summer. Do you get a totally different poem or do you get a refinement of the original poem?

The answer is yes to both because you can ask it to use the initial output as a basis or you can tell it to completely recreate another one. That ongoing prompting will do exactly as I've asked it but sometimes it's easier to forget that it's not a human being on the other side of the computer and it will give you exactly what you've asked for. Once again the accurate speaking is very important but in the same token, especially in my earlier encounters with GPT, I'm a horror for a typo. I was amazed that it would still understand what I was saying, but that's probsaid, having those mistakes in it probably limited the quality of the results I got. It was very good at understanding mispelled text and I think this will become commonplace when computers have an ear and a voice so we can talk to models like GPT rather than type things. I think this would be convenient but because of the nature of the accuracy required, unless you're very good at saying exactly what do you need, sometimes it's easier to type it out and edit it before you fire it across GPT's bows, so to speak. 95% of the prompts I create I usually create in an editor first and I refine it and change it until I'm happy it says what I need it to say, then paste it into the prompt box and ask GPT to act on it.

I guess the hesitation for us having been trained in the world of the search engine where we had a small space where we were putting in our questions or our queries, with GPT of its various versions, we can have it essentially as long as we want, the instructions that we provide for it.

There are limits and those limits are escalating. At the time of recording, the limit of what I can put in as a query to the model I use, is 4,096 tokens. I've just dropped in another term there. A token is how GPT breaks down the messages we ask it into small, almost like syllables. It's probably equivalent to about 1500 - 2000 words.

That's a fair size instruction!

That said, sometimes you might want to summarise an article. It's the classic example of "too

long didn't read" (TLDR). You can grab text and paste it in and say, please summarise this for me or please place it in bullet point form. This is something I use it for a lot. It saves me from reading volumes and volumes of text but gives me the parts I need to know, gives me the salient points and it's very very good at that. But once again, with longer articles, by pasting them in you will hit the limit of what it will accept as a maximum amount of text. That problem's been partially solved by the fact that we can ask it to look at a webpage and summarise that. That's getting a little bit away from what we're talking about.

In the first part of our discussion, you did briefly touch on the idea that people might lose jobs, jobs might be at risk and that the best thing to do in twerms of help with that, in terms of your understanding, was to try it out and become conversant with it or to find a way to use it that would enhance your productivity or your work environment. If we look at the productivity first of all and I'll put you on the spot and say can we talk about your productivity, so in terms of productivity gains that you've seen over the time you've been using it, has it been worthwhile? Given that it's something you have to learn to use.

I think the bottom of the learning curve is quite short and I think you'll find after you start using it and get comfortable with it, you definitely will improve your productivity. In the beginning I used a term that I learned from a gentleman by the name of Dean Jackson and he talked about efficiency as how many hours per hour as opposed to kilometres or miles per hour. My hours per hour when I really started to use the model as well as I could, I gauged myself as running at about 10 hours per hour. Now that wouldn't be something I could keep up on a continual basis but it will certainly give you a multiplier and in some instances you might find you get efficiencies that are far greater than that depending on the sort of work that you are doing. As a result of that people using these technologies will be far more efficient. The down side of that is that we will need less people to do the same work.

It might be that it gives you the opportunity to offer different services that you didn't have the human resources available for. All of those projects that might have been on the backburner because we haven't got the resources, the physical resources, to investigate them, to carry out, to implement them, now we're freeing up some human resources to work on those projects, whether it's a combination of using their skills with GPT to develop a project to a stage where it needs to be tried out, so all that research, the trial phase and the investigation and groundwork that happens with any sort of project, if you've got someone who has got time freed up from their other activities because of their proficient use of GPT, they can then go on to a project that has been on the backburner for a while and get it started. In that same sense, if you are using these "extra" resources, because what it provides is extra resources, that can be applied to any activity.

I think that's true. I tend to look at GPT as being my incredibly intelligent, incredibly smart assistant.

That's it for part two of our discussion with Jamie Wadley about all things GPT. Join us for part 3, for now, I'm Kim Baillie, she's Fulyana Orsborn and this is Inside Exec.