Welcome to the next lesson in this Introduction to Prompt Engineering. Now we're going to talk about how to get high and then higher quality results from your prompts and your interactions with AI.

So this is a callback to an image that we looked at at the beginning. Again, the most important thing is figuring out what point B is and you're using AI to accelerate progress or the process of getting there. What you'll notice here is there are some arrows that are looping back to the beginning or to different parts of the process as you get from point A to point B.

When it comes to getting high quality or high enough quality results from your prompts and then higher and higher quality results on the engineering side when you start to go for reliability and generalizability of whatever it is that you're building, you want to really use the scientific method, which is to say you have a hypothesis of if I do this, it'll get me to point B. You collect experimental data by trying it and just seeing what happens. And then you make sense of that data and you incorporate your findings into iterative improvements. So it's in some sense, it's pretty simple, but that's how you initially get good enough results and then move on to the next step of the process or improve reliability and so on and so forth.

So prompt iteration, it really is a simple process. One. Define the problem and figure out point B.

Two. Try to solve the problem with extreme constraints and AI assistance. Talked about how hyper growth mindset is really useful in getting the most out of this technology: so ambitious goal, extreme constraints, do it fast, do it now, see if you can get there.

And then assess the results with an acceptable threshold. This is again, another reason why defining the problem is so important. If you know what it is that makes for a quality output or actually gets a particular result, you'll be much better at assessing that result and iteratively improving what it is you're doing with AI.

So you want to assess the result. Figure out what that acceptable threshold is in terms of whatever it is you're going to publish, send. Does the code that you write, if you're writing software, actually do the thing you want to do? That's kind of easy to figure out. So that's a good example.

And then if the answer is no, you go back to one or two. Ideally two, or maybe you need to redefine the problem in the first place. And that's pretty much it. So you're just trying different things and you're making it work.

And then if I would say after generally speaking, three attempts, you can't get directly from point A to point B, then you probably need to break the problem into smaller pieces. Funny enough, this is the mirror image of telling the model. Let's think this through step by step. So maybe you need to step. Think it through step by step, right?

So break it down into those incremental pieces and then figure out, “Okay, if let's say it's, I think it's four steps, this must be true. Then this must be true. This must be true. This must be true.” Or those outputs feed into each other's inputs. For example, if you're going to write a LinkedIn post about really salient topics or something like that, just asking the model or in your industry or something, right? We're going to talk about enterprise AI. You can ask the model, “Okay, write the post.”

Maybe the post is not particularly good. It's not your voice. All this kind of stuff. Then you take a step back. Want this to be higher quality so why don't you ask me questions about generative AI? Then it gets a little bit better. You answer the questions. You turn them into a post and you're like, “E, this is still generic. It's not super up to date.”

Then you say, “Okay, I'm going to break this into something even earlier, which is to, I want you to take the most recent law informed YouTube video transcript on a particular topic, Generative AI. Then come up with questions based on that.” And that may work. And then you get slightly higher quality results 'cause you're gonna provide answers that are about a salient topics or things that are top of mind. And salient just means new and recent.

And maybe that even that's not good enough. You have to break it down further and say, “I want you to actually find the most interesting parts of the discussion in the video. I'm gonna tell you which ones I like. We're going to turn those into questions, et cetera.”

So just through that process, you're trying to get to a result. In that case, the bar for quality is quite high so you're going to have to break it down in order to get to where you want to go. But that's basically it, right?

Define the problem. Try it. Assess the results. Iterate. And if you can't get to a good enough result, keep breaking it down until you're happy with those incremental steps.

Okay. In defining the problem, it's super important that you think hard about the problem that you're trying to solve. If this is not clear, I would actually recommend going for a walk. Just step away from your computer. Try to think through what it is you're trying to do. Maybe for 10-15 minutes, just think in silence. And then pull out your phone. And then for maybe another 5-10 minutes, open the ChatGPT app, ramble into it, and have ChatGPT try to bring a little bit more structure to help you figure out what it is you're trying to do. You're like, “Oh, I'm trying to do this. Activate this particular marketing channel or I've never posted something on Instagram reels. I don't know how this works.” And it'll explain some stuff to you and you figure it out. And then you come back maybe to your desk or wherever your tools are and you can start incorporating that into something that's a little bit more directed where you can delegate tasks. Right?

So AI as a thought partner is the way I like to put it, but you really want to make sure you put the time and energy into defining the problem first. Then you try to attempt to solve the problem. So, ambitious goal. Do things quickly. Use AI to get there. Apply some extreme constraints, especially in terms of time. And just do it. Just generate the experimental data.

And then you want to assess quality of the results. I would say, generally speaking, as a sort of rule of thumb, if the results are >= 50% of what you need and you saved 50 to 90% of your time getting the result, this is going to be different depending on the task. But you're trying to figure out acceptable trade off between, let's say, like quality and cost especially quality and time, right?

Because you're trying to get things done faster. Then at that point, maybe you're ready to engineer the prompt for better performance or incorporate some of the techniques we talked about before retrieval, augmented generation, all these kinds of things to make it better.

And if it's less than 50% of where you want it to be, and it's just not workable at all, you probably need to break it down into those intermediary steps. So, if you get it good enough, similar to if an analyst does a first draft of something, you're like, “Okay, this is good enough structurally. This is right. I can just put some finishing touches on it and move forward.” That's one of one of two possible outcomes.

The other being, “No, this is just not good. I need to do basically a page one rewrite, which means I need to break the problem down into smaller steps.” So as you're doing the experimentation, that's something to keep in mind. You're making this assessment, like, “is this good enough?” “Is it not good enough?” “ If it is good enough, then improve quality incrementally. And if it's below that line, there's something structurally messed up that needs to get fixed. All right.

So coming back to this awesome image from the tweet from Andre Karpathy, he's over at OpenAI, talked about earlier. I added a few of these recursive arrows. When it comes to task accuracy, quality, it's again, this process of trying something, generating the data, assessing the data, figuring out if it's good enough, and then moving forward. And if you're able to, you know, get here by simply providing a command, zero shot prompt means you just don't provide any examples of the output that you want.

Then it might be the case that you're ready to go to the next level. You provide some of those examples. If those are just not working and you're not able to boost the quality enough that you can move to the next level, maybe there's something fundamentally messed up with like your original prompt, which means that could be improved. So better instructions, more context, a better goal, or you need to break it down into pieces.

But this is more applicable, by the way, to just like a single prompt. So good instruction. Then you provide some gold standard examples. Then you keep trying to provide more and more contextual information, either in the original prompt or through discourse.

I mentioned this previously. If you engage with the medium, if it's a one off thing where you're not just building something where you want to fit lots of the same inputs in and have a highly engineered prompt that may turn into software at some point, then you can have a conversation and bring in context where it's appropriate until again, you get to the highest levels that you can really push it to where it's acceptable and you can move on to the next task. I'm going to skip over this.

There is this technique called fine tuning, where once you have enough examples, let's say roughly a hundred is the current threshold, right? If you're doing lead scoring, you have a hundred gold standard examples you can work with. You can actually engage in a process called fine tuning, where you show the mode those examples. And it will update the actual structure of its neurons. It's almost like the way that we, once we get enough repetitions and we develop like a certain intuition around how to make a snap judgment or whatever it is, you can do that at a certain point to eke out those last few points of reliability. We're not going to talk about how to do this in practice. It's still relatively early days for making this work practically, but just know that is available to you. This isn't so different than from teaching somebody how to do something, right? So you give them kind of the instructions and say, this is what I want to get done.

You show them some examples of really good work that they use for reference. And then as they work through a problem, or even at the beginning, you just provide lots of context and that's going to make the quality go up. And then finally, at some point in the future, they've developed some mastery, some expertise to the point that they become almost like unconsciously competent, right?

Like they're making good predictions and they may not necessarily even know why, but they've just seen this so many times that they're able to.

So that's how you get high quality results. And again, this idea that you're using the scientific method to develop a hypothesis to test it, to make sense of the data and then keep moving forward is very important.

That's how you get high and then higher quality results through prompt engineering.

I'll see you in the next lesson.