Creating Advanced prompts

Let's recap some learnings from the previous chapter:

> Prompt *engineering* is the process by which we **guide the model towards more relevant responses** by providing more useful instructions or context.

There are also two steps to writing prompts, constructing the prompt, by providing relevant context and the second part is *optimization*, how to gradually improve the prompt.

At this point, we have some basic understanding of how to write prompts, but we need to go deeper. In this chapter, you will go from trying out various prompts to understanding why one prompt is better than another. You will learn how to construct prompts following some basic techniques that can be applied to any LLM.

Introduction

In this chapter, we will cover the following topics:

- Extend your knowledge of prompt engineering by applying different techniques to your prompts.
- Configuring your prompts to vary the output.

Learning goals

After completing this lesson, you'll be able to:

- Apply prompt engineering techniques that improve the outcome of your prompts.
- Perform prompting that is either varied or deterministic.

Prompt engineering

Prompt engineering is the process of creating prompts that will produce the desired outcome. There's more to prompt engineering than just writing a text prompt. Prompt engineering is not an engineering discipline, it's more a set of techniques that you can apply to get the desired outcome.

An example of a prompt

Let's take a basic prompt like this one:

> Generate 10 questions on geography.

In this prompt, you are actually applying a set of different prompt techniques.

Let's break this down.

- **Context**, you specify it should be about "geography".
- **Limiting the output**, you want no more than 10 questions.

Limitations of simple prompting

You may or may not get the desired outcome. You will get your questions generated, but geography is a big topic and you may not get what you want to due the following reasons:

- **Big topic**, you don't know if it's going to be about countries, capitals, rivers and so on.
- **Format**, what if you wanted the questions to be formatted in a certain way?

As you can see, there's a lot to consider when creating prompts.

So far, we've seen a simple prompt example, but generative AI is capable of much more to help people in a variety of roles and industries. Let's explore some basic techniques next.

Techniques for prompting

There are some basic techniques that we can use to prompt an LLM. Let's explore them.

- **Zero-shot prompting**, this is the most basic form of prompting. It's a single prompt requesting a response from the LLM based solely on its training data.
- **Few-shot prompting**, this type of prompting guides the LLM by providing 1 or more examples it can rely on to generate its response.
- **Chain-of-thought**, this type of prompting tells the LLM how to break down a problem into steps.
- **Generated knowledge**, to improve the response of a prompt, you can provide generated facts or knowledge additionally to your prompt.

- **Least to most**, like chain-of-thought, this technique is about breaking down a problem into a series of steps and then ask these steps to be performed in order.
- **Self-refine**, this technique is about critiquing the LLM's output and then asking it to improve.
- **Maieutic prompting**. What you want here is to ensure the LLM answer is correct and you ask it to explain various parts of the answer. This is a form of self-refine.

Vary your output

LLMs are nondeterministic by nature, meaning that you will get different results each time you run the same prompt.

Running the same prompt again generates a slightly different response:

> So is the varied output a problem?

Depends on what you're trying to do. If you want a specific response then it's a problem. If you're ok with a varied output like "Generate any 3 questions on geography", then it's not a problem.

Using temperature to vary your output

Ok, so we've decided we want to limit the output to be more predictable, that is more deterministic. How do we do that?

Temperature is a value between 0 and 1, where 0 is the most deterministic and 1 is the most varied. The default value is 0.7. Let's see what happens with two runs of the same prompt with temperature set to 0.1:

As you can see, the results couldn't be more varied.

> Note, that there are more parameters you can change to vary the output, like top-k, top-p, repetition penalty, length penalty and diversity penalty but these are outside the scope of this curriculum.

Good practices

- **Specify context**. Context matters, the more you can specify like domain, topic, etc. the better.
- Limit the output. If you want a specific number of items or a specific length, specify it.
- **Specify both what and how**. Remember to mention both what you want and how you want it, for example "Create a Python Web API with routes products and customers, divide it into 3 files".
- **Use templates**. Often, you will want to enrich your prompts with data from your company. Use templates to do this. Templates can have variables that you replace with actual data.
- **Spell correctly**. LLMs might provide you with a correct response, but if you spell correctly, you will get a better response.