# Prompt Engineering Exercises

2023-11-17

## 1 Exercise: Create a Basic Prompt

Using the general prompting template provided in the slides, create a prompt that asks the language model to act a personal tutor for you. Make the model explain to you what a transformer is. Specify the:

- ROLE,
- TASK_DESCRIPTION,
- TASK_SPECIFICATION,
- and FORMAT_OUTPUT.

### 1.1 try some variations

Try experimenting with some variations of this yourself. Change the role, task, and output format. How does the model respond to these changes?

*You should notice that the model responds strongly to how you describe each of the 4 basic prompt structures.*

## 2 Exercise: Order of Operations

### 2.1 Answer first

Give ChatGPT the task to multiply 3 numbers together, which are at least 3 digits in length. Specify the output such that ChatGPT first gives you the result of the multiplication, and only then it gives you the 3 numbers. What do you see happening?

*The model starts by generating a number, and only then it is allowed to think of the 3 numbers it needs to multiply to get there. This is a difficult problem for the LLM and it often fails. Since the new patch, it tends to ignore the structure requirements for this question.*

### 2.2 Answer last

Do the same, but now reverse the order. Make sure ChatGPT first gives you the 3 numbers, and then make it calculate the multiplication. Is anything different, why?

*Now the answer makes more sense, but it often still fails. ChatGPT, and most LLMs at the moment, are not good at math and should not be used for math.*

## 3 Exercise: Order of Operations + room to think

### 3.1 Answer first

Let the model check the answer below. First, only allow the model to answer with the verdict correct or incorrect. Then ask for the explanation in the next message.

```
f"""
Determine if my answer below is correct.

I'm trying to figure out how expensive my prompting of the GPT4 API will be in total.
I'm going to run around 1000 prompts. For these prompts, the average prompt-lengt is 250 w
The average response length of the model for these prompts is 300 words.

The API pricing is as follows:
$0.03/1k prompt tokens.
$0.06/1k sampled tokens.

We can assume that 1 token is on average 3/4th of a single word.

I found the following answer:
18 dollars.
"""
```

*Most of the time, this results in a faulty answer, but not always.*

### 3.2 Answer last

Now do the same, but instruct the model to first work through the problem itself. Ask it to **think step by step**.

*When doings this, the answers tend to be beter, although it might still make mistakes.*

### 3.3 Reflection

Why would it help language models to work step-by-step?

*It gives the model room to think. It is allowed to spend a lot more compute on the same problem, which helps getting more reasonable results.*

## 4 Exercise: Spelling

1. Ask the model the reverse the word: "congratulations".
2. Ask the model to reverse the word and spell it out letter-by-letter. What do you see? Why do you think this happens.

*Due to the tokenized nature of the model, it has no good grasp of spelling and letter order inside tokens.*

### 4.1 A closer look

Use the openai-tokenizer to look at the words? What do you see?

*You see that the word is divided into multiple tokens.*

### 4.2 Self-reflection

Ask the model whether the answer it gave was correct, what happens? Why do you think this is the case?

*The funny thing is that you can often get these models to correct themselves In this case it might help because just asking the model whether it is wrong might nudge it in the direction to realize that the spelling was faulty*

#### 4.2.1 Self-reflection on the API-costs questions

If the answer given was incorrect, see how it responds if you ask it to reflect on the answer given.

## 5 (Optional) Few-shot prompting + Chain-of-thoughts

One simple but time consuming task you might often do is to take notes during a meeting, then type these out into a mail to send to the team. However, how you take notes is very specific to you, just like how you want and need the mail to look.

We can guide the model by giving it a few examples of our task, this is called *Few-shot prompting.* Also, I want you to take into consideration the order of operations and how to go from meeting notes to this email.

**Please don't send any private or proprietary information to third parties.**
You can use your notes for this Masterclass if you like.

## 6 (Optional) Generating movie reviews

One of the areas where these models really shine is content generation. Or at least, getting you started with written content.

For one of the later exercises we will use these models to analyse movie reviews. In preperation for this exercise, have the model generate a review of a movie of your own choice. Make use of the following rules:

- from the review, the sentiment regarding the movie should be slightly ambigous.
- Make clear what the 'reviewer' liked about the movie, and what it didn't like.
- Begin the movie review with some general information about the movie: such as the director, star actors/actresses, runtime and other such information.
- Make sure the model mentions at least 3 other related movies in the review.
- Don't include a star rating or other numeric rating within this review.
- The review should be around 500 words long.