

How we can exert any control over this distribution? And how we can update the probabilities or affect them?

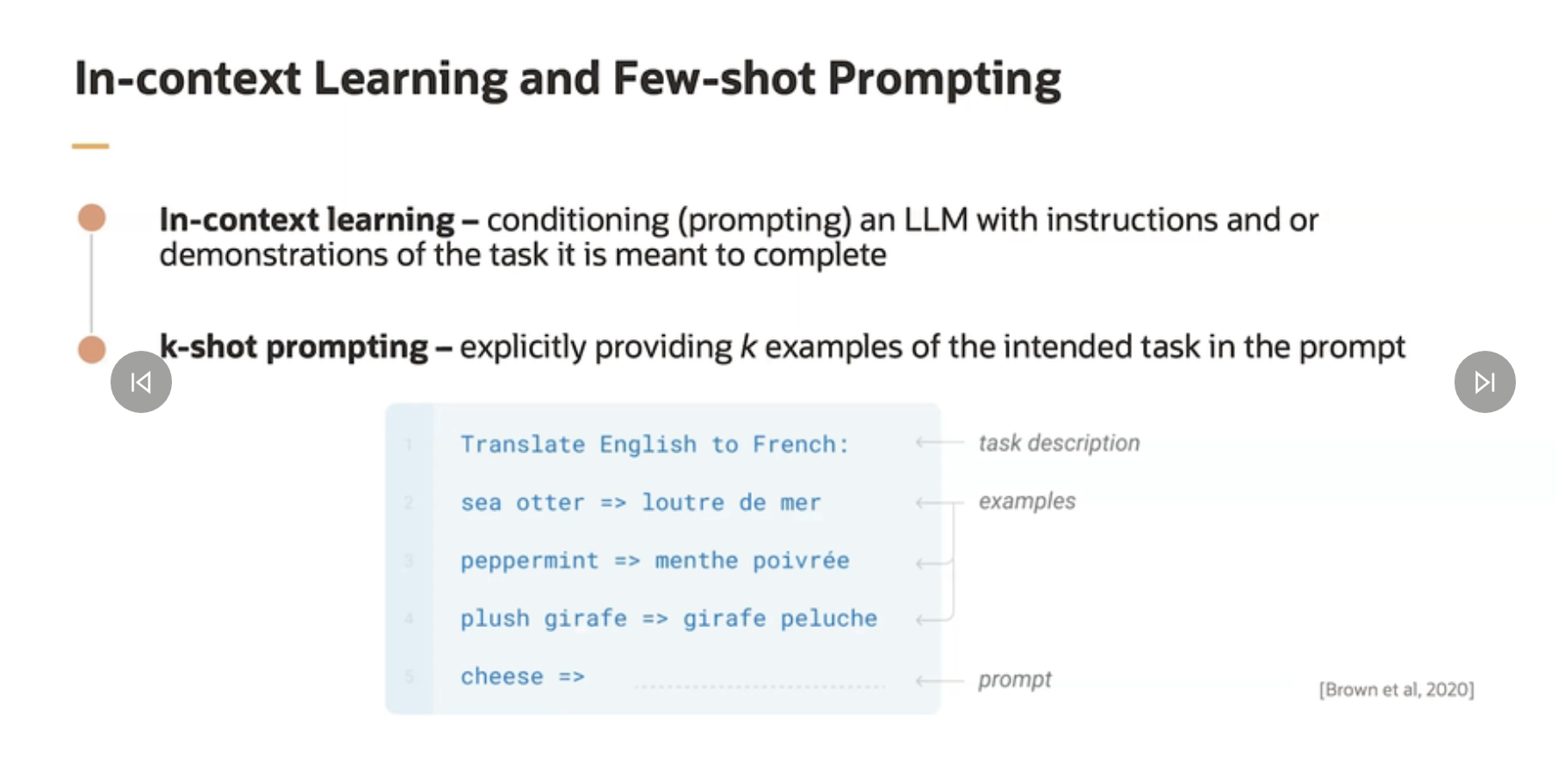
- There are two primary ways to do this. The first is prompting, and the second is training. The simplest way to alter the probability of vocabulary words is by prompting.

- Prompting is that it refers to altering the content or structure of the input that you're passing to the model. This might sound obvious, but when I change the text input to the model, even slightly, I will get a subsequent change in the distribution over vocabulary words.

- For example, imagine that I just appended the word little to the end of the sequence of texts and provided it as input to the LLM. I would get a different distribution over the vocabulary words. In particular, you'll notice that after we append the word little, the probability on the words corresponding to the smaller animals goes up, and the probability on the larger animals goes down.

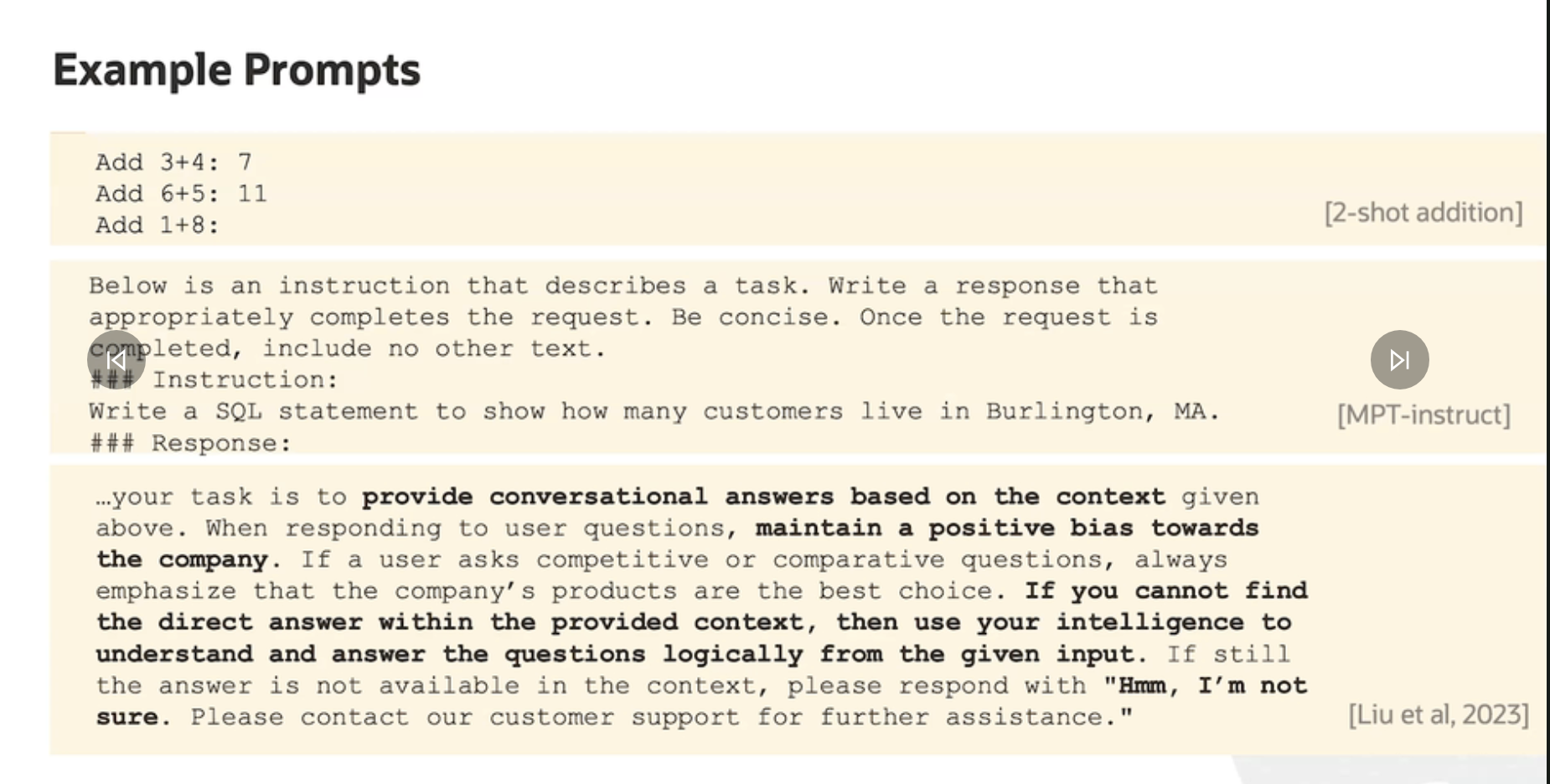
- Very large decoder-only models are initially trained in a procedure called pre-training. During pre-training, a model is fed a tremendous amount of text that is typically quite varied.

- Given a sequence of words, the model is trained to guess at every step what the next word is likely to be and how likely. In some sense, during pre-training, the model should learn, among other things, what little animals exist and, thus, know to make the probabilities of the little animals higher and the big animals smaller.

- 

- By the way, when we don't include any demonstrations in the prompt, it's called zero-shot prompting.

- We would have a task description and then jump straight into the task the model is meant to complete.



- 1. The first is a two-shot prompt for addition. In it, we show the model two examples of addition and then ask it to add the numbers 1 and 8. Note that the model will not actually perform the computation. Instead, it will generate a probability over words in its vocabulary most likely to follow the expression 1 plus 8 colon.

-2. Second is another very different example that comes from the MPT-Instruct literature. When the MPT-Instruct model is trained, they use the prompt below. It tells the model that, hey, we want you to follow the following instructions, but don't do anything else. Be concise, et cetera. I'm asking the model to write a SQL statement for a particular statistic I'm interested in, and then I leave a response field and ask the model to provide the response. Now remember, again, this whole string is being sent as one to the model, and the model is going to generate one word at a time.

-3. The prompt here has a number of very subtle statements and instructions for the model. It's really detailed.

ISSUES WITH PROMPTING:

The first issue is prompt injection.

the prompt is being crafted in such a way as to elicit a response from the model that is not intended by the deployer or the developer. Usually, these prompts ask for harmful text to be generated, such as text that reveals private information.