

Few-shot prompting

1. Few-shot prompting

00:00 - 00:08

Welcome back! In this video, we'll take our LLM prompting skills up a level and master few-shot prompting.

2. Limitations of standard prompt templates

00:08 - 00:35

So far, we've used `PromptTemplate` and `ChatPromptTemplate` to create reusable templates for different prompt inputs. These classes are great for handling prompts containing a single example, but they don't scale well if we need to integrate many examples from a dataset. The `FewShotPromptTemplate` class allows us to convert datasets like these into prompt templates to provide more context to the model.

3. Building an example set

00:35 - 00:50

Let's say we have a list of dictionaries containing questions and answers. If we have another data structure, like a pandas `DataFrame`, there's usually a simple transformation to get to this point, like the `.to_dict()` method in this case.

4. Formatting the examples

00:50 - 01:17

Before we dive in with creating the few-shot prompt template, we need to decide how we want to structure the examples for the model. We create the prompt template, using the `PromptTemplate` class we've used before, to specify how the questions and answers should be formatted. Invoking this template with an example question and answer, we can see the "Question" prefix was added, and a new line was inserted.

5. FewShotPromptTemplate

01:17 - 01:38

Now to put everything together! `FewShotPromptTemplate` takes the examples list of dictionaries we created, and the template for formatting the examples. Additionally, we can provide a suffix, which is used to format the user input, and specify what variable the user input will be assigned to. Let's see our prompt in action!

6. Invoking the few-shot prompt template

01:38 - 01:57

We invoke the prompt template with an example user input, and extract the text from the resulting prompt. The prompt correctly appended `Question:` to the user input, displaying it

underneath the example questions and answers. Now let's test that this prompt template is actually functional in an LLM chain.

7. Integration with a chain

01:57 - 02:17

We instantiate our model, and chain the prompt template and model together using the pipe operator from LCEL. The model response can be extract from the response object via the `.content` attribute, which shows the model was able to use the context provided in our few-shot prompt.

8. Let's practice!

02:17 - 02:20

Now it's your turn!