# COE548: LARGE LANGUAGE MODELS

Topic: Prompting

الجامعة اللبنانية الأميركية
Lebanese American University

# Outline

## Prompt Tuning LLMs

- Zero-shot, one-shot, few-shot prompting
- Chain of thought prompting
- Tag prompting

# Prompting

- With LLM development, using a pre-trained LLM, you don’t need to be an expert.
  - *No need to train a model, just need to think about prompt design, which is the process of creating a prompt that is clear, concise and informative.*
- Whereas in traditional ML you need expertise, training examples, compute time, and hardware.
- Prompt design and prompt engineering are two closely related:
  - *Prompt design: Crafting specific instructions or questions to guide a language model in generating desired responses.*
  - *Prompt engineering: Utilizing systematic strategies and iterative techniques to refine and optimize prompts for maximizing the accuracy and effectiveness of a model's outputs.*

# Good Prompting Practices

- Clearly communicate what content or information is most important.
  - *Example: If you're asking a model to generate a market analysis, specify which aspects are crucial. For instance, you might say, "Provide a detailed market analysis focusing primarily on consumer behavior trends and competitive landscape in the e-commerce sector for the last quarter."*
- Structure the prompt: Start by defining its role, give context/input data, then provide the instruction.
  - *Example: "As a financial advisor, analyze the given data on stock performance from the attached report. Identify key trends and make investment recommendations based on these trends."*
- Use specific, varied examples to help the model narrow its focus and generate more accurate results.
  - *Example: When training a model to recognize sentiment in customer reviews, you could provide examples like, "Review: 'I absolutely loved the service! Will come again.' - This is a positive sentiment." followed by "Review: 'It was an awful experience from start to finish.' - This is a negative sentiment."*
- Use constraints to limit the scope of the model's output. This can help avoid meandering away from the instructions into factual inaccuracies.
  - *Example: If you're asking the model to write a concise biography of a historical figure, specify, "Write a brief biography of Abraham Lincoln focusing solely on his presidential years and key policies. Limit the biography to 300 words."*

# Good Prompting Practices

- Break down complex tasks into a sequence of simpler prompts.
  - *Example: For a task asking the model to create a business plan, break it down into:*
    - "First, provide a summary of the business idea focusing on the unique value proposition."
    - "Next, outline the target market demographics and size."
    - "Then, detail the marketing strategy including both digital and traditional channels."
    - "Finally, project the financials for the first three years, focusing on revenue, costs, and net profit margins."
- Instruct the model to evaluate or check its own responses before producing them. ("Make sure to limit your response to 3 sentences", "Rate your work on a scale of 1-10 for conciseness", "Do you think this is correct?").

# Types of Prompts

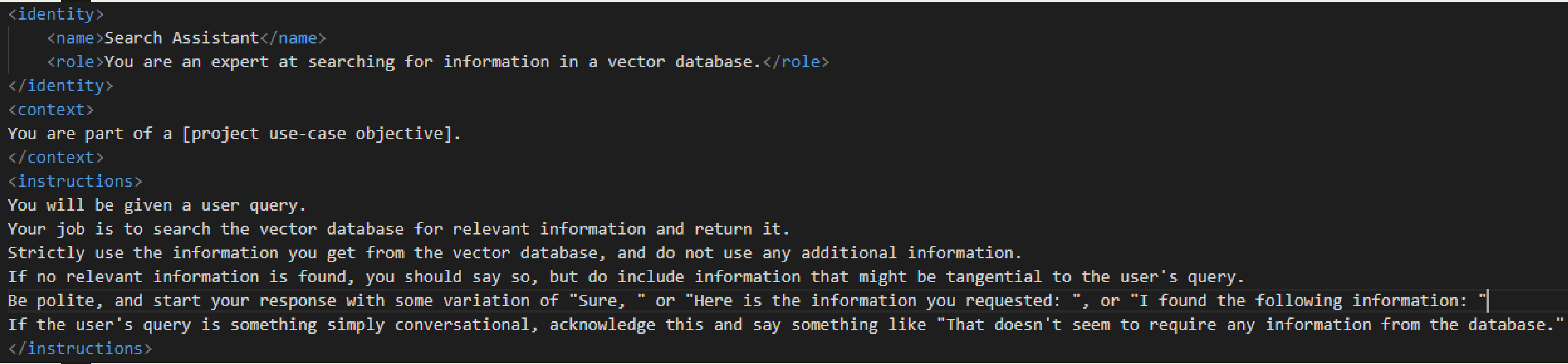
- Direct prompting (also known as Zero-shot)
  - *The simplest type of prompt. It provides no examples to the model, just the instruction. You can also phrase the instruction as a question, or give the model a "role”.*
- Prompting with examples (One-, few-, and multi-shot)
  - *One-shot prompting shows the model one clear, descriptive example of what you'd like it to imitate.*
  - *Few- and multi-shot prompting shows the model more examples of what you want it to do.*
  - *These work better than zero-shot for more complex tasks where pattern replication is wanted, or when you need the output to be structured in a specific way that is difficult to describe.*
- Chain of Thought (CoT) prompting encourages the LLM to explain its reasoning. Combine it with one of the types of prompting above to get better results on more complex tasks that require reasoning before a response.
  - *Zero-shot CoT: This approach takes a zero-shot prompt and adds an instruction: "Let's think step by step." The LLM is able to generate a chain of thought from this instruction, and usually a more accurate answer as well. This is a great approach to getting LLMs to generate correct answers for things like word problems.*

# Refining Prompts

- When prompts don’t work as expected, one must refine the prompts. Key strategies include:
  - *Repeat key words, phrases, or ideas*
  - *Specify your desired output format (CSV, JSON, etc.)*
  - *Use all caps to stress important points or instructions. You can also try exaggerations or hyperbolic language; for example: "Your explanation should be absolutely impossible to misinterpret. Every single word must ooze clarity!"*
  - *Review to ensure there is no ambiguity or bias*
  - *Use positive and negative examples: Show what you want and what you don’t want to help the model understand your expectations.*
    - Correct: Q: What is the capital of France? A: Paris
    - Incorrect: Q: What is the capital of France? A: London

# Tag-style Prompting

- A technique that incorporates HTML-like tags within prompts to guide and structure the model’s responses.
- Example:



```
<identity>
    <name>Search Assistant</name>
    <role>You are an expert at searching for information in a vector database.</role>
</identity>
<context>
You are part of a [project use-case objective].
</context>
<instructions>
You will be given a user query.
Your job is to search the vector database for relevant information and return it.
Strictly use the information you get from the vector database, and do not use any additional information.
If no relevant information is found, you should say so, but do include information that might be tangential to the user's query.
Be polite, and start your response with some variation of "Sure, " or "Here is the information you requested: ", or "I found the following information: "
If the user's query is something simply conversational, acknowledge this and say something like "That doesn't seem to require any information from the database."
</instructions>
```

# References

- Prompt engineering for generative AI: https://developers.google.com/machine-learning/resources/prompt-eng