Natural Language Processing
Large Language Models Usage and Evaluation Patterns

Felipe Bravo-Marquez

December 4, 2023

Introduction

- Since the inception of Large Language Models, various patterns of use and evaluation of this technology have emerged.
- In this talk, we will try to organize these patterns and give a general overview of them.

Source:

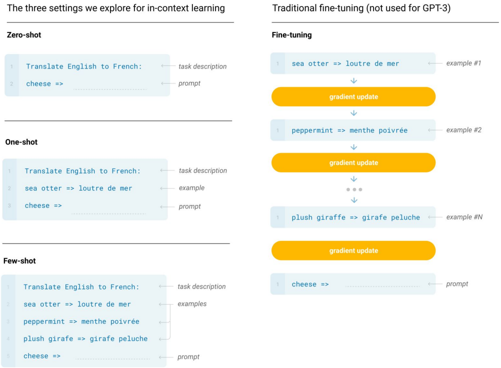
`https://www.masayume.it/img/masayume/Large-Language-Models.jpg`

Recap: What is an LLM

- Large Language Model: An autoregressive language model trained with a Transformer neural network on a large corpus (hundreds of bullions of tokens) and a large parameter space (billions) to predict the next word.
- It is usually later aligned to work as a user assistant using techniques such as Reinforcement Learning From Human Feedback [Ouyang et al., 2022] or supervised fine-tuning.
- Some are private (access via API or web browser): Google Bard, ChatGPT, etc.
- Others are open (model's weights can be downloaded): Llama, LLama2, Falcon, etc.

Zero-shot, One-shot, and Few-shot Learning

The most remarkable feature of these models is their few-shot, one-shot, zero-shot learning capabilities (also known as “in-context-learning”).


The three settings we explore for in-context learning
Zero-shot
Translate English to French:
task description
cheese =>
prompt
One-shot
Translate English to French:
task description
sea otter => loutre de mer
example
cheese =>
prompt
Few-shot
Translate English to French:
task description
sea otter => loutre de mer
examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
prompt
Traditional fine-tuning (not used for GPT-3)
Fine-tuning
sea otter => loutre de mer
example #1
gradient update
peppermint => menthe poivrée
example #2
gradient update
plush giraffe => girafe peluche
example #N
gradient update
cheese =>
prompt

This means that they can learn new tasks without large amounts of human-annotated data.

Talk Overview

- Despite the recency of this technology, its adoption has been tremendous in many areas.
- Below, we propose a simple categorization of the ways in which LLMs are used and evaluated.
- These patterns will serve as the narrative backbone of this presentation.

Usage Patterns

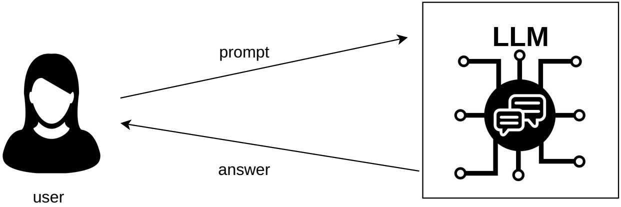
1. General-domain Assistant
2. Domain-specific Assistant
   - 2.1 Retrieval-augmented generation
   - 2.2 Fine-Tuning
3. LLM-based Applications
   - 3.1 API calls
   - 3.2 Autonomous Agents

Evaluation Patterns

- MTBench
- LLM Arena

Usage Pattern 1: General-domain Assistant

- In this pattern a user interacts with the LLM proving prompts as input and receiving a text as answer.
- The knowledge the LLM has access is limited to the corpus on which it was trained and the context given in the prompt.



Tasks

LLMs can solve many tasks with this pattern:

- Textual: Language understanding and common sense (e.g., rewriting, summarizing, translating, answering questions)
- Arithmetic: Mathematical reasoning (it can fail in many cases though)
- Visual: Multimodal reasoning involving pictures (GPT-4, Llava)
- Symbolic: Structured input such as programming languages

Source:

`https://twitter.com/IntuitMachine/status/1727079666001870877`.

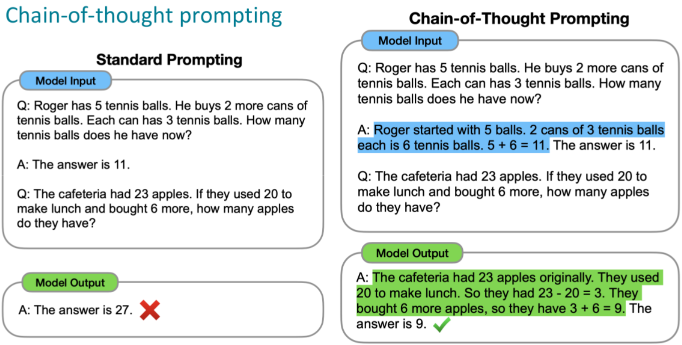
Prompt Engineering: Guiding the Language Model

Prompt engineering, often referred to as “Prompting,” is the discipline or “art” of crafting effective prompts to guide the Language Model (LM) towards generating accurate responses. Some common prompting guidelines:

- **Clarity and Conciseness**: Clearly articulate the prompt to minimize ambiguity and ensure the LM understands the task at hand.
- **Use of Specific Examples**: Provide concrete examples within the prompt to offer the LM context.
- **Role-based Prompts**: incorporating roles into the prompts (e.g., a tour guide, a teacher, a doctor).
- **Desired Output Specification**: Clearly define the desired format of the output (e.g, JSON, HTML, csv).

Chain-of-thought Prompting

- Chain-of-thought prompting is a simple mechanism for eliciting multi-step reasoning behavior in large language models.
- This method involves augmenting each exemplar in a few-shot prompt with a connected sequence of thoughts, creating a structured chain of logical steps. [Wei et al., 2022]



Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

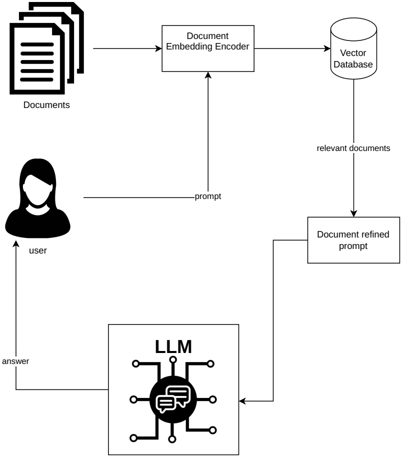
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Usage Pattern 2: Domain-specific Assistant

- Idea: Incorporate domain-specific knowledge not covered in training (e.g., recent news, private documents).
- This is very common for companies developing chatbots with private documents or for creating more domain-specific chatbots.
- There are two main patterns to achieve this:
  1. Retrieval-Augmented Generation (Vector Databases)
  2. Fine-Tuning

Usage Pattern 2.1: Retrieval-Augmented Generation

Idea: Incorporate domain-specific knowledge into the query using information retrieval and document embeddings (i.e., densely encoded vectors that capture the semantic information of the document). [Lewis et al., 2021].

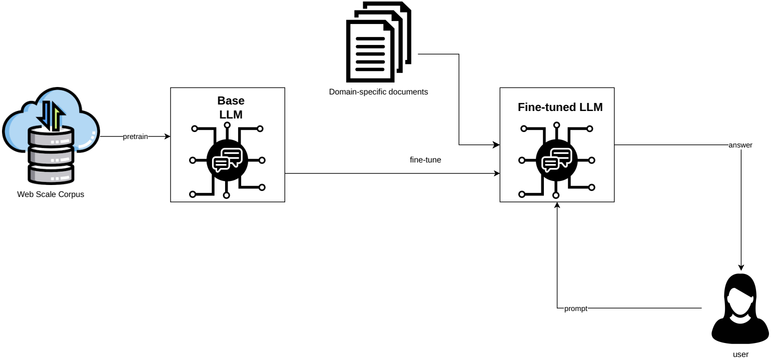


Retrieval-Augmented Generation Process

1. Encode all domain-specific documents using document embeddings and store them in a vector database.
    - OpenAI provides a vectorizer called (text2vec-openai), but there are many open source alternatives.
    - There are also many vector databases available, a popular one is **weaviate**.
2. Encode the prompt with the same vectorizer used to encode documents.
3. Use prompt embedding and the vector database to retrieve relevant documents based on similarity.
4. Create a refined prompt that includes a domain-specific role, the user prompt, and the retrieved documents as contexts.
5. Send the refined prompt to the LLM and return the response to the user.

Usage Pattern 2.2: Fine-Tuning

Idea: Incorporate domain-specific knowledge by fine-tuning a pre-trained LLM with the next-token prediction task over a domain-specific corpus and interact with the resulting LLM.



This can be computationally expensive unless some tricks are used.

Instruction Fine-Tuning

- Idea: instead of training the LM with raw text with next token prediction, train it with pairs of prompts and user-aligned answers.
- Paid Fine-Tuning (GPT-4??)
- OpenAI offers many more specific gpts: `https://openai.com/blog/introducing-gpts`
- Alpaca, Vicuna, Llama, Llama2
- https://blog.gopenai.com/paper-review-qlora-efficient-finetuning-of-quantized-llms-a3c857cd0cca

Datasets for Instruction Fine-Tuning

- Standford Alpaca Dataset (Vicuna)
- ShareGPT (Alpaca)
- Dolly-15K
- Orca Dataset

Parameter Efficient Fine Tuning

- Lora, QLora
- https://blog.gopenai.com/paper-review-qlora-efficient-finetuning-of-quantized-llms-a3c857cd0cca

Token-Incrementation

- Lora, QLora
- https://blog.gopenai.com/paper-review-qlora-efficient-finetuning-of-quantized-llms-a3c857cd0cca

Applications

- LLMs can be embedded into any software via API calls. For example, a search engine (you.com)
- https://gptstore.ai/
- For example, PDF summarization software. You write software that first converts the PDF to raw text and then sends it to an LLM for summarization.
- Or a software for summarizing videoconferences. First the audio is transcribed and then summarized with an LLM.

Autonomous Agents

- Agents are a special kind of LLMs application in which the LLM serves as the reasoning and planning component of the software.
- agent in the sense of perceiving an environment and taking actions to achieve goals.

LLMBench and LLm Arena

- Standard NLP evaluation: human annotated gold-labels and metrics.
- LLMS are intrinsically multi-task and not easily evaluated with this approach.
- Machines evaluating machines??
- MT-bench (categories)
- HuggingFace Open LLM Leaderboard
- LLM Arena

Questions?

Thanks for your Attention!

References I

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2021).
Retrieval-augmented generation for knowledge-intensive nlp tasks.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022).
Training language models to follow instructions with human feedback.
Advances in Neural Information Processing Systems, 35:27730–27744.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. (2022).
Chain of thought prompting elicits reasoning in large language models.
arXiv preprint arXiv:2201.11903.