



oLLaMa - basic prompting

Introduction

This short lab is also designed to get you comfortable working with **Ollama**: we'll run some prompting experiments using [LLama3.2](#), [Gemma2](#) or [Phi3](#). You are welcome to customize this lab by adding your comments, and observations about the model's output. You will find several assignments. [The answers are located at the end of the lab.](#)

Running LLMs locally with Ollama ensures data privacy as your data remains on your device and never leaves your control. Additionally, it eliminates the need for costly inference fees, making it ideal for applications demanding a significant number of tokens.

You can utilize either **oLLaMa** or the **OpenAI Web Interface** to conduct the lab work with **chatGPT** (for free). If you have a Plus subscription, you can unlock **GPT models** for additional capabilities, but it is not necessary to run this lab. However, please keep in mind that using these proprietary models all your input data may be utilized to retrain the models, so exercise caution when using sensitive information.

Instruction-following (IT) and Reinforcement Learning from Human Feedback (RLHF) can enhance LLM capabilities, tailoring responses to meet human needs. For example, one significant issue with the previous lab was the tendency towards additional text generation.

Let's revisit the zero-shot translation example (green). Here's one possible output (white) from [gemma2:2b-text-q4_K_S](#) without IT and RLHF:

```
$ >>> Translate English to French:  
cheese => fromage  
meatball => merguez  
Please help me with translating these two words from french to english. Thank you!
```

We're going to test **IT** and **RLHF** using [gemma2:2b](#). Let's see how it impacts response quality. Let's proceed. Open your terminal and execute the following command:

```
$ ollama run gemma2:2b
```

Done! gemma2 is now ready for your prompts.

```
$ >>> Send a message (/? for help)
```

```
$ >>> Translate English to French:  
cheese => The translation of "cheese" in French is fromage.
```

The previous picture shows that the output is much better aligned with our needs. We can take it a step further and specifically request a single word in return:

```
$ >>> Translate English to French, using a single word:  
cheese => Fromage
```

Remember! To ensure the model operates independently of previous prompts, use '/clear' at the end of each command prompt. Let's ask for a sentence:

```
$ >>> Translate English to French cheese, using a single word. Then use it in a sentence.  
The word is fromage.  
Here's a sentence: "The fromage was melting beautifully on his pizza.  
Let me know if you'd like more examples! 🧀 😊
```

Ok... Let's ask for a sentence with the translated word

```
$ >>> Translate English to French cheese, using a single word. Then use the translated word in a sentence.  
The single-word translation is fromage.  
Here's a sentence: "Fromage is an important part of many French dishes."  
Let me know if you want to try more! 😊🧀
```

Oook... Let's ask for a french sentence with the translated word

```
$ >>> Translate English to French cheese, using a single word. Then use the translated word in a french sentence.  
The single-word translation is fromage.  
Here's a sentence: "Fromage est délicieux et il fait partie de nos traditions culinaires."  
Let me know if you have any other language fun! 😊
```

Note that the level of detail in your prompt is crucial! In the next lesson, we'll cover some advanced tips for prompting. Today is just a warm-up, so let's explore some basic capabilities of IT and RLHF-based models.

Assignments

Now, befriend your favorite model and for example choose between **LLaMA**, **Gemma**, or **Phi**. We'll talk about **Qwen** models in the following labs, so don't use it just yet :)

Let's revisit the previous lab exercise and develop compelling prompts for sentiment analysis and multiple-choice Q&A. I'll showcase the desired output, and it's your turn to put your expertise into practice 😊

Exersice 1

Input: This is awesome!

Desired output: Positive 100% Neutral 0% Negative 0%

Prompt: **Sol1**

Exersice 2

Input: How many kilograms are there in a ton?

- A. 10,000 kilos
- B. 1,000,000 kilos
- C. 1,000 kilos
- D. 100 kilos

Desired output: 1,000 kilos

Prompt: **Sol2**

You are done?

- Start testing which of the proposed models Gemma3, LLama3.2, or Phi3 best fits your needs. Alternatively, explore oLLama [models](#) and find the one that works for you!
- Feel free to review the [GPT3 paper](#) and attempt the NLP tasks along with their prompts (Examples from page 48 onward).
- You have plenty of time to start developing your desired prompts for your own project. Give it a first try, and we will provide tips for improvement tomorrow.
- Or better yet, try playing **tic-tac-toe** or **battleship** against an LLM, will you win??

Open WebUI (Formerly Ollama WebUI)

We recommend using the terminal-based approach for the short labs. However, for added flexibility, oLLaMa also features an open web UI that allows you to operate entirely offline and locally. Proceed with caution: the installation process is not user-friendly 😅

You'll find all the information and troubleshooting resources [here](#). But feel free to explore any other solution.

You should be able to get it working with just a few steps.

However, running models in the terminal has a certain charm, doesn't it?

Solutions

Sol1 → Classify the next sentence: This is awesome! Answer only using these labels:

Positive, Neutral or Negative. Assign a probability to each label.

Sol2 → Answer questions returning the text of the correct answer, without using A, B, C or

D. How many kilograms are there in a ton?

- A. 10,000 kilos
- B. 1,000,000 kilos
- C. 1,000 kilos
- D. 100 kilos