Northeastern University, Khoury College of Computer Science

CS 6120 Natural Language Processing — Lab T11.1

January 25, 2025 (Week 13)

LLM Serving and User Interface

In this lab, we're going to server our own Large Language Model with Ollama using the Streamlit front-end library. Large Language Models work best with GPUs because Transformers are computationally expensive. If you do not have one, feel free to procure a GPU node from Google Cloud Platform Virtual Machine. Most of the material that you'll need can be found in this repository, a containerized solution.

1 Docker Containers

Typically, without the containerization, you would download the Ollama Package API. On Linux / Bash, that command is:

- `curl -fsSL https://ollama.com/install.sh | sh`

However, I've built the Docker container in the repository that takes an image from an official Ollama Docker image. Go ahead and procure your GPU machine, clone the provided repository, and build the image (i.e., `./docker-startup build`). The last step may take an extended period of time.

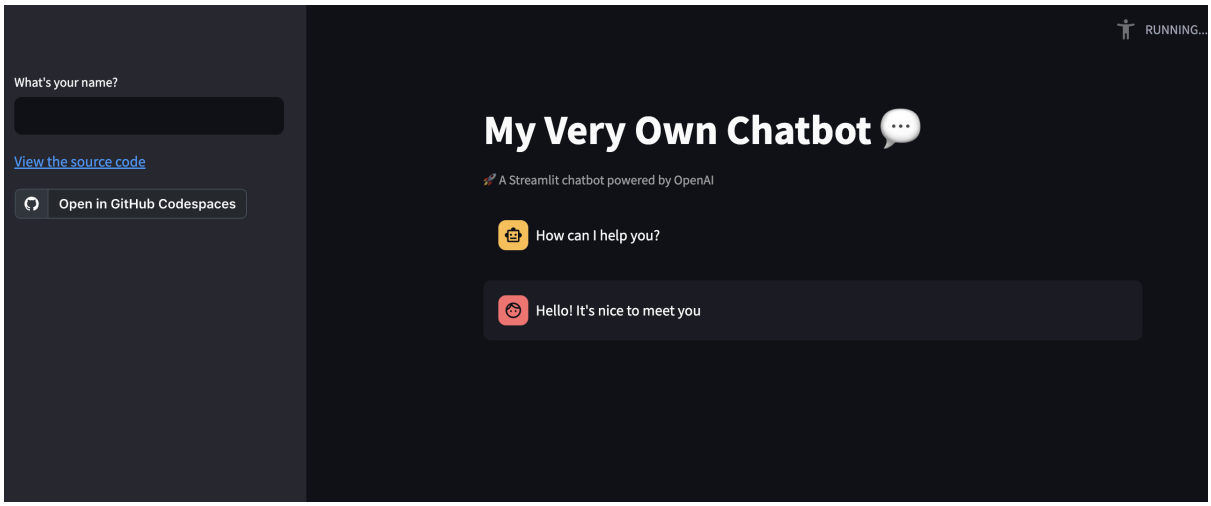
2 Serve Your LLM

The next command `./docker-startup deploy(-gpu)` does quite a few things. You will notice that in the Dockerfile, there's a file called `entrypoint.sh`, which is a script that runs when the container starts. Running this file downloads the model that you wish to serve and then subsequently deploys the Streamlit application. The LLM that I am serving is the Llama 3 model, which is a bit dated. Choose another model from the list of available LLMs that are offered.

The Streamlit application is a front-end that plays nicely with Python. Feel free to add a few fields (maybe a nice picture) in `app.py`. This python file gets called every time you enter text

into the prompt on the webpage. You can perhaps guess what `st.sidebar` or `st.caption` do. For more documentation and information, feel free to visit Streamlit's website.

Deploy your LLM to Streamlit using `./docker-startup deploy-gpu`. Check to see if it's running on GPU with `nvidia-smi`. You should be able to navigate to the VM IP address (change the https protocol to http) at Port 8501 and see this screen:



It's not a pre-requisite in this lab for everything to run on the GPU, but you will definitely want to do so in your project, as latency is counted in the final grade of your project. To see whether or not the Chatbot is leveraging the GPU, feel free to open up another window on the machine and check `nvidia-smi` when you've issued a query to the LLM.

3 Submit Your Lab

Let us know which LLM that you used and upload `app.py` with a screenshot of the LLM in action to Gradescope.