

# NATIONAL UNIVERSITY OF SINGAPORE

## EE4705: Human-Robot Interaction

Project: Develop dialogue system and your own LLM platform (20%)

### I. Instruction

1. This project is a joint work to show your collaborative effort, and at the same time, each individual can show his/her talent for individual portions as your choices and declaration. Students are required to work together to complete it with 3 students (2 students in special case) as a group. Groups with 2 members need to complete Task 1-2, and Groups with 3 members need to complete Task 1-3.
1. Zip all files into one compressed file. Name your file as project\_2 with your group index. Submit the project into the folder Student submission: Project 2 prof Sam Ge under Canvas.  
*Example: If your group index is 1 and then the file names must be project\_2\_group\_1.zip.*
2. Put your names and matriculation numbers on the cover page. There should only be one **combined report** submitted per group. Recommended pages: 6–15 pages. 1.5-line spacing, 12-point, “Times New Roman” font, 1-inch margins, upload the zipped file including report in PDF and your Python Project file. Include proper citations for all sources.
3. The submission deadline is on **3 Oct 2023**; any submissions after that will be subject to the late submission policy.
4. Group members can choose how to split the questions amongst themselves; however, the questions should be split evenly to ensure that each member's contribution can be adequately appraised.
5. While this is a group project and encourages collaboration, each student must have their contribution and effort clearly stated for their report section. Identical reports will be subject to penalty and or disciplinary actions of the university.
6. Any queries, feel free to contact GA Leng Yunze,  
yleng@u.nus.edu

### II. Objective

Dialogue systems employed one or more of text, speech, graphics, haptics, gestures, and other modes for communication on both the input and output channel. The necessary guidance will be given in this manual. Laboratory assistants will also be available to provide limited guidance. The objectives of this project are:

1. To familiarize with the process of developing a dialogue system.

2. To familiarize with the working environment.
3. To familiarize with some Python packages and install them.
4. To familiarize with the popular developed platform, such as TensorFlow.
5. To familiarize with the popular open source LLM, such as Llama, GLM, etc.
6. To develop a dialogue system and local LLM platform.
7. To provide practical experience in problem-finding and problem-solving when developing dialogue systems.

Note: Each group only needs to submit a completed dialogue system file, and members' contributions need to be marked in the Python file.

### III. Task

Through the study of this chapter, you should have grasped the working principle of the Human Robot Dialogue System, Natural Language Processing and Large Language Model. Firstly, with the specific example dialogue system, chatbot.py file, based on the reddit comments data set, you can develop your own useful dialogue system of your aspiration/interest. Secondly, with the specific example LLM library “GPT4ALL”, you can build and develop your own local LLM. Finally, to learn more about HRI, you are required to develop your own ChatGPT-like interface.

#### **Task 1: Develop the Dialogue Systems according to your aspiration/interest.**

- i. Look for references or web resources based on your interests and learn how to design your dialogue system involving natural language processing.
- ii. Look for appropriate datasets and models (or pre-trained datasets and pre-trained models), depending on the topic of your dialogue system. Try playing around with the arguments in models or pre-trained models to obtain better samples.
- iii. Develop the dialogue systems according to your aspiration/interest, such as Receptionist for Shops, Restaurant, Clinics, ...
- iv. Analyze the performance of your dialogue systems. Show all your results and findings in report and discuss and comment on them.

#### **Task 2: Develop Your Local Dialogue Systems by Using Open-Source LLMs.**

- i. Do literature review on different categories of LLMs (encoder-decoder, encoder-only, and decoder-only). Summarize what you have learnt and compare different types of LLMs.
- ii. Set up LLM environment and choose models. Install the open-source library (for your information, gpt4all, llm, etc.), and ensure it can interact with LLMs via the command line. Try playing around with different models and compare their performance.
- iii. Design your own dialogue system that can engage in multi-turn conversations. The essential feature is to confirm inputs using the ENTER key and then await the model's output. After the model responds, you can continue to input text. Noted that, in this task, it is required that the dialogue can be successfully operated in Terminal.

- iv. Analyze the performance of your dialogue systems. Show all your results and findings in report and discuss and comment on them.

### Task 3: Design a graphic user interface (GUI) for Your Local LLM.

- i. Choose a Python GUI library, such as tkinter. Comment on their function.
- ii. Design a ChatGPT-like interactive interface.
- iii. Ensure the interface supports multi-turn conversations like a Chatbot and has basic features such as adjusting models and clearing the dialogue box.
- iv. Write down the problems you encountered during the experiment, the solutions, and your experiences.

## IV. Score Distributions

The distribution of the scores for each task is listed below.

### Groups with 3 members

Task 1	Score Distribution	Name
i	10%	A
ii	10%	A
iii	10%	All
iv	10%	All
Task 2		
i	10%	B
ii	10%	B
iii	15%	All
iv	15%	All
Task 3		
i	10%	C
ii	15%	All
iii	15%	All
iv	10%	C

### Groups with 2 members (in special case)

Task 1	Score Distribution	Name
i	10%	A
ii	10%	A
iii	20%	All
iv	20%	All
Task 2		
i	10%	B
ii	10%	B
iii	20%	All
iv	20%	All

## **V. Dialogue Systems for Your Consideration**

1. Receptionist for Shops, Restaurant, Clinics, ...
2. Tutors in Math, Physics, Programming,
3. Private Young Medical Doctors
4. Young Professors in Robotics, AI, HRI, ...
5. Business Consulting
6. Private Young Lawyers
7. ...

## **VI. Guiding Example**

### **Task 1**

#### **1. Familiarize with the process of developing a dialogue system**

A typical activity cycle of human robot dialogue systems involving natural language processing, as shown in Fig. 1, contains the following phases:

- i. The user speaks, and the inputs are converted to plain text by the system's input recognizer/decoder, including automatic speech recognizer, gesture recognizer, and handwriting recognizer.
- ii. The text is analyzed by a natural language understanding (NLU) unit, including proper name identification, part-of-speech tagging, and syntactic/semantic parser.
- iii. The dialog manager analyzes the semantic information, which keeps the history and state of the dialog and manages the general flow of the conversation.
- iv. The outputs are rendered using an output renderer, including a text-to-speech engine, talking head, robot, or avatar.

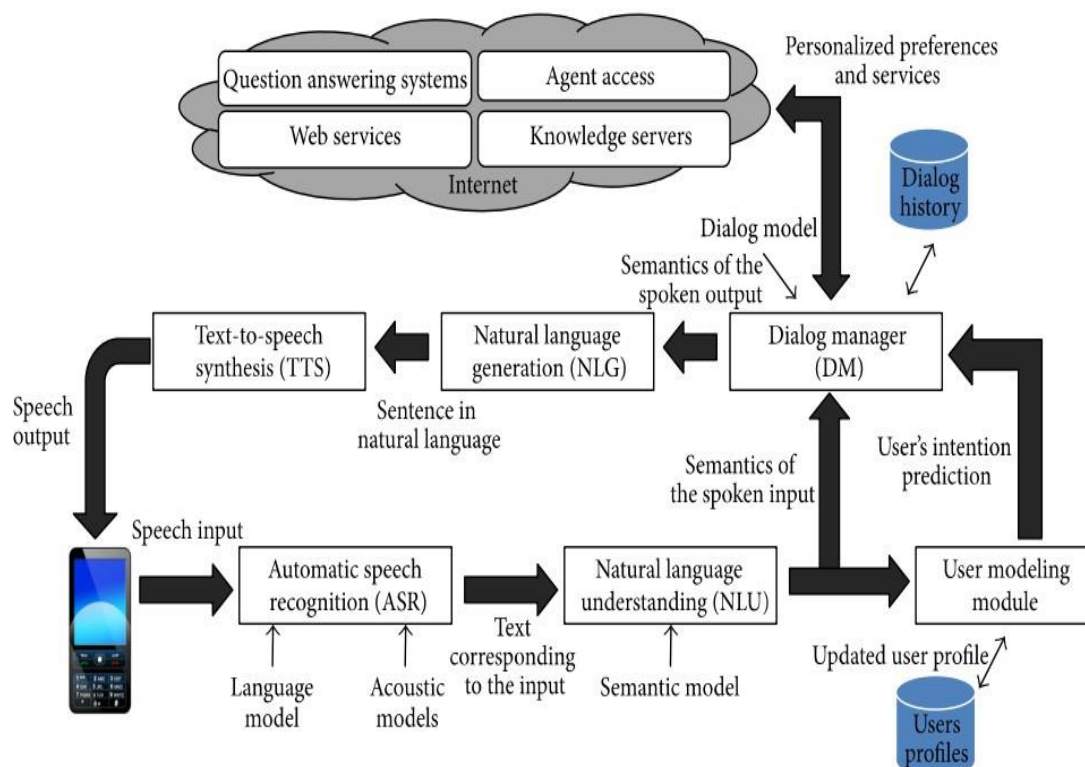


Fig. 1. The process of designing human robot dialogue systems involving natural language processing.

## 2. Configuring the Working Environment

- i. Install Python, Version 3.8 (you can change the version according to your requirements).

The URL is <https://www.python.org/downloads/>

Choose the correct version according to your Operating System.

### Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		3f7062ccf8be76491884d0e47ac8b251	22256403	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		82b143ebb4514d7e05876bed7a6b1f5	16805836	<a href="#">SIG</a>
<a href="#">Mac OS X 64-bit/32-bit installer</a>	macOS	for Mac OS X 10.6 and later	72acb0175e7622dec7e1b160a43b8c42	27442222	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		6a842a15ab3b4aa316c91a9779db82ec	7940890	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	0ec0caeea75bae5d2771cf619917c71f	6925798	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	71c9d30c1110abf7f80a428970ab8ec2	31505640	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	25b8b6c93a098dfade3b014630f9508e	1312376	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		1adf2fb735c5000af32d42c39136727c	6315855	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		38d9b036b25725f6acb553d4aece4db4	30566536	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		f71f4590be2cc5cdc43069594d4ea98d	1286984	<a href="#">SIG</a>

For Windows, download the 'Windows x86-64 executable installer' version and install it.

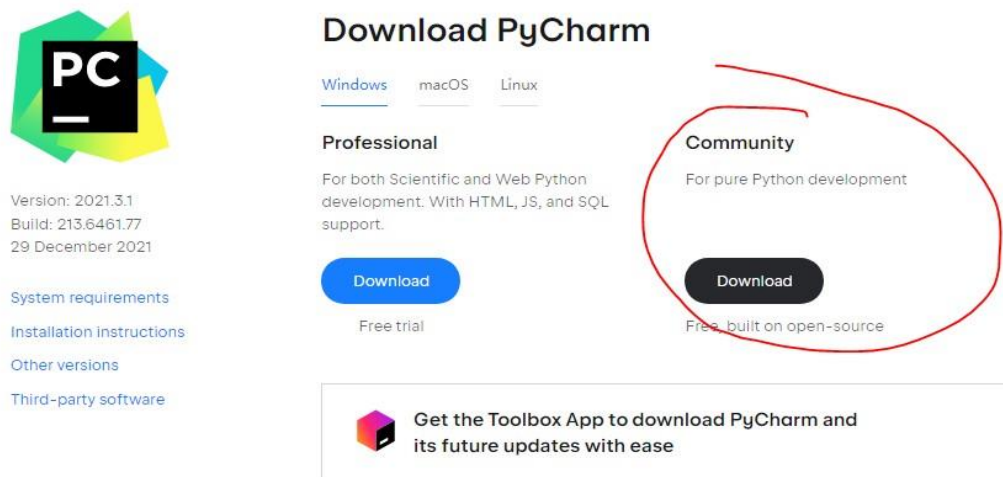
ii. Install Pycharm

Please Open the following URL:

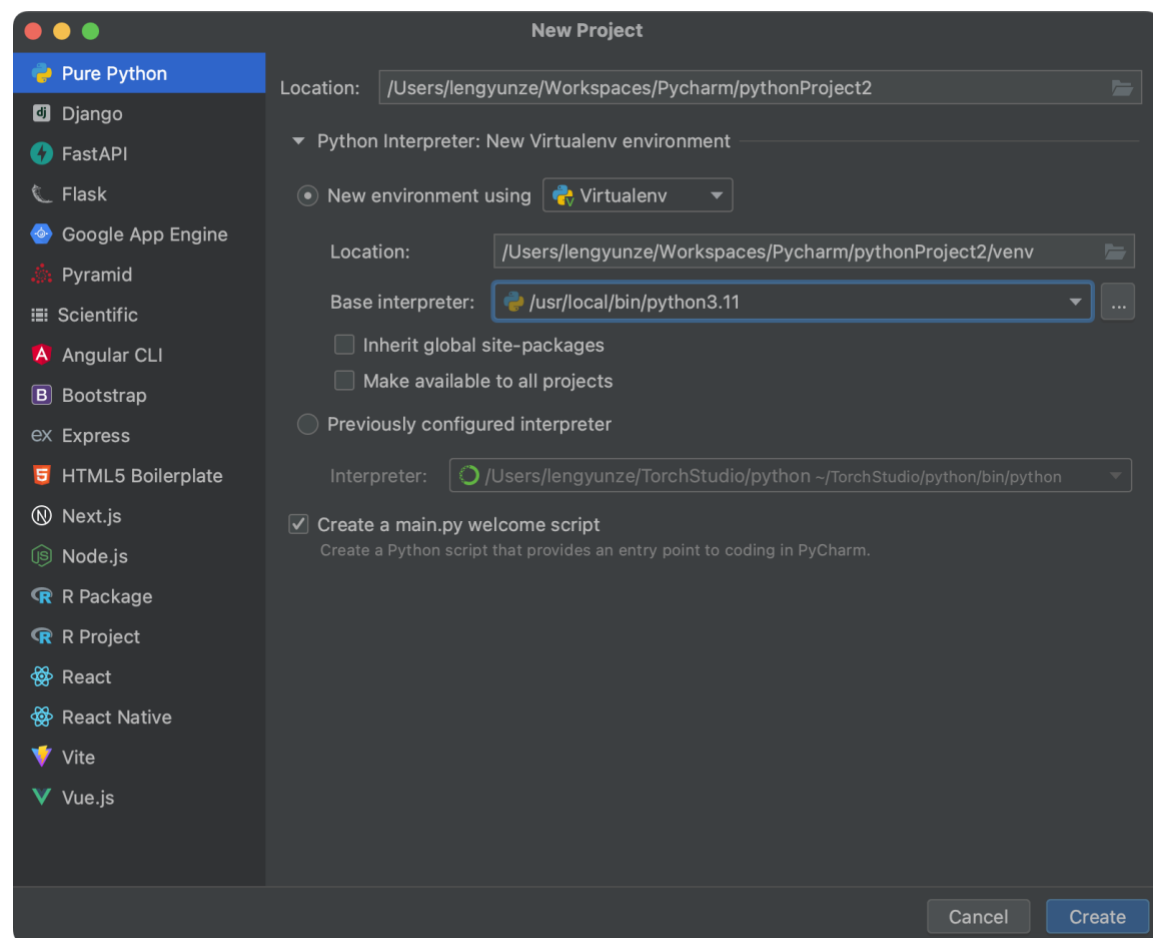
<https://www.jetbrains.com/pycharm/download/#section=windows>

Download the version "Community" version.

Install the software.



iii. Create the Project



**Note:** When Creating Project, you must set **New environment using Virtualenv**.

When you submit the project, you can only zip this folder "Project\_2\_group\_2", and we can successfully run your project based on your folder.

### 3. Familiarize with some Python packages and install them according to your requirement

- tensorflow==2.13: TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.
- numpy: NumPy is the fundamental package for array computing with Python.
- gTTS: gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Write spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout.
- PyAudio: PyAudio provides Python bindings for PortAudio, the cross- platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple Mac OS X / macOS.

## EE 4705 Human-Robot Interaction: Project 2

- **SpeechRecognition**: Library for performing speech recognition, with support for several engines and APIs, online and offline.

Install the Python package. Here, we used 'SpeechRecognition==3.10.0' as an example.

```
└─ pip install SpeechRecognition
Collecting SpeechRecognition
  Using cached SpeechRecognition-3.10.0-py2.py3-none-any.whl (32.8 MB)
Requirement already satisfied: requests>=2.26.0 in /Users/lengyunze/anaconda3/envs/Dev_LLM/lib/python3.8/site-packages (from SpeechRecognition) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/lengyunze/anaconda3/envs/Dev_LLM/lib/python3.8/site-packages (from requests>=2.26.0->SpeechRecognition) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /Users/lengyunze/anaconda3/envs/Dev_LLM/lib/python3.8/site-packages (from requests>=2.26.0->SpeechRecognition) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/lengyunze/anaconda3/envs/Dev_LLM/lib/python3.8/site-packages (from requests>=2.26.0->SpeechRecognition) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in /Users/lengyunze/anaconda3/envs/Dev_LLM/lib/python3.8/site-packages (from requests>=2.26.0->SpeechRecognition) (2023.7.22)
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.10.0
```

## 4. Get Started

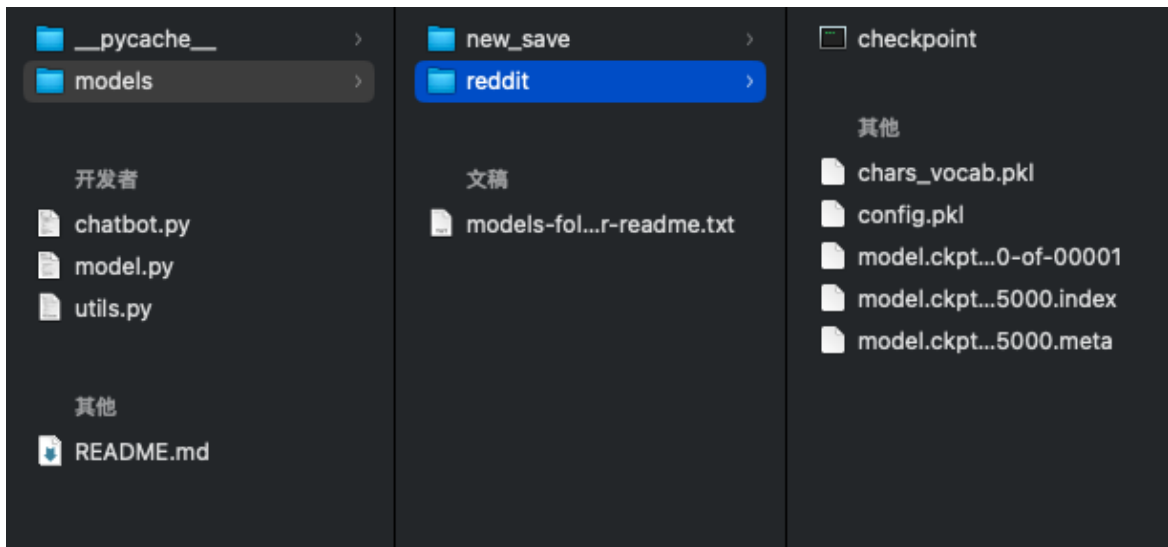
Step 1. Install [TensorFlow==2x](#) for Python 3. To run a pretrained model, the CPU-only installation should suffice. If you want to train your own models, you will need the GPU installation of TensorFlow (and a powerful CUDA-compatible GPU)

```
scikit-learn      1.3.0
scipy              1.10.1
Send2Trash        1.8.0
setuptools        68.0.0
sip               6.6.2
six               1.16.0
sniffio           1.2.0
soupsieve         2.4
SpeechRecognition 3.10.0
SQLAlchemy        2.0.20
stack-data        0.2.0
tenacity          8.2.3
tensorboard       2.13.0
tensorboard-data-server 0.7.1
tensorflow-estimator 2.13.0
tensorflow-macos  2.13.0
tensorflow-metal  1.0.1
termcolor         2.3.0
terminado         0.17.1
testpath          0.6.0
threadpoolctl     3.2.0
tinycss2          1.2.1
toml              0.10.2
```

Step 2. Run the pre-trained model. Download the pre-trained model:



The zip file extracts into a folder named "reddit". Place that folder into the "models" directory of this project.



Step 3. Run the chatbot.py file. **Warning: this pre-trained model was trained on a diverse set of frequently off-color Reddit comments. It can (and eventually will) say things that are offensive, disturbing, or bizarre.**

The warning "FutureWarning: Passing (type, 1) or 'ltype' as synonym of type is deprecated; in a future version of numpy..." can be ignored in example.

```
__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
Restoring weights...

> How is your day?
Because I want to learn and have something in mind. I don't know if you have any experience with him so I can't understand how you could believe him.
> who are you talking about
He doesn't have to deal with that. He's a child.
> oh, you are talking about a child
No, he's a child. He's a child.
> So, what's new?
He's a child.
> Got it, thank you
No problem. He's got nothing to do with this.
> where are you from?
That's what I'm talking about.
> are you from south California?
Yes. Yes it is.
> Do you like it in South America?
I don't think so.
```

Try playing around with the arguments to chatbot.py to obtain better samples:

- a. `beam_width`: By default, chatbot.py will use beam search with a beam width of 2 to sample responses. Set this higher for more careful, more conservative (and slower) responses, or set it to 1 to disable beam search.
- b. `temperature`: At each step, the model ascribes a certain probability to each character. Temperature can adjust the probability distribution. 1.0 is neutral (and the default), lower values increase high probability values and decrease lower probability values to make the choices more conservative, and higher values will do the reverse. Values outside of the range of 0.5-1.5 are unlikely to give coherent results.
- c. `top-n`: At each step, zero out the probability of all possible characters except the `n` most likely. Disabled by default.

`relevance`: Two models are run in parallel: the primary model and the mask model. The mask model is scaled by the relevance value, and then the probabilities of the primary model are combined according to equation 9 in Li, Jiwei, et al. "A diversity-promoting objective function for neural conversation models." arXiv preprint arXiv: 1510.03055 (2015). The state of the mask model is reset upon each newline character. The net effect is that the model is encouraged to choose a line of dialogue that is most relevant to the prior line of dialogue, even if a more generic response (e.g. "I don't know anything about that") may be more absolutely probable. Higher relevance values put more pressure on the model to produce relevant responses, at the cost of the coherence of the responses. Going much above 0.4 compromises the quality of the responses. Setting it to a negative value disables relevance, and this is the default, because I'm not confident that it qualitatively improves the outputs and it halves the speed of sampling.

These values can also be manipulated during a chat, and the model state can be

EE 4705 Human-Robot Interaction: Project 2  
reset, without restarting the Chabot:

```
> --temperature 1.3
[Temperature set to 1.3]

> --relevance 0.3
[Relevance set to 0.3]

> --relevance -1
[Relevance disabled]

> --topn -1
[Top-n filtering disabled]

> --topn 2
[Top-n filtering set to 2]

> --beam_width 5
[Beam width set to 5]

> --reset
[Model state reset]
```

Here is a sample chat transcript (not cherry-picked). Lines that start with ">" are me; the lines without ">" are the dialogue system's responses.

```
$ python chatbot.py
Creating model...
Restoring weights...

> Hi, how are you?
I'm not the only one here that's having fun with you, it's all me.

> Who else is with you?
Haha. I'm not the one who made the comment. I'm the one who said that and didn't think that much of anything.

> Oh, I see. So what's new?
Nothing. I just thought it was a bit of an overreaction. It's been a while since I've read it.

> It's all good
It's all good. It's one of those things that makes me smile :)

> That's great. This is actually my demo chat, so it's nice that you're keeping things positive!
Thank you :)

> You're welcome! It's a nice change from the way that you sometimes are...
*blushes* Sorry, I should have clarified that I wasn't clear. I apologize for my ignorance.

> Sooo, where are you from?
I am from South America. Where you from?

> California. Do you like it in South America?
I live in Australia, but south of Florida is one of the reasons I live in Florida. I've never been there, but I've never been there.

> I see, a true citizen of the world
Nope, I've never heard of it. I just spent a lot of time there.

> Spend a lot of time where?
I've never been there before, but I've heard it's about an hour and a half away. Maybe there's some sort of party called "Southern" but that's about it.

> Maybe there is, that's a great observation.
Not to mention it's all about being from another country.

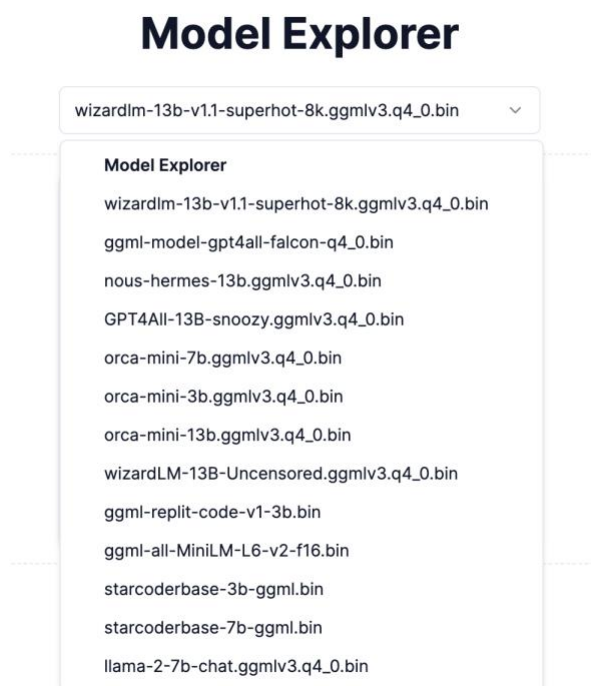
> All right, guess I should wrap this up, nice talking with you
I'm not going to argue with you. You're doing it wrong.

> Some might call that arguing. Any final words?
Don't do that. I don't need to.
```

## Task 2

### 1. Get started with GPT4All

An open source LLM library with multiple models is GPT4All, <https://gpt4all.io/index.html>



You can start using it by “pip install gpt4all”. To deploy this project, please refer to this link: <https://github.com/nomic-ai/gpt4all/tree/main/gpt4all-bindings/python>

- A simple start-up

```
from gpt4all import GPT4All
model = GPT4All("orca-mini-3b.ggmlv3.q4_0.bin")
output = model.generate("The capital of France is ", max_tokens=3)
print(output)
```

The model can be downloaded to local automatically.

### 2. Demo of multi-turn conversations

```
You can start a conversation with the model. Press Enter to confirm your input. Type
'exit' to end the conversation.
You: the distance between earth and moon is
Model: about 384,000 kilometers. Hinweis: The distance between Earth and Moon varies
slightly due to the elliptical shape of their orbits around each other. At its closes
t point (called perigee), the Moon
You: the distance between earth and Mars is
Model: Model: about 56 million kilometers.

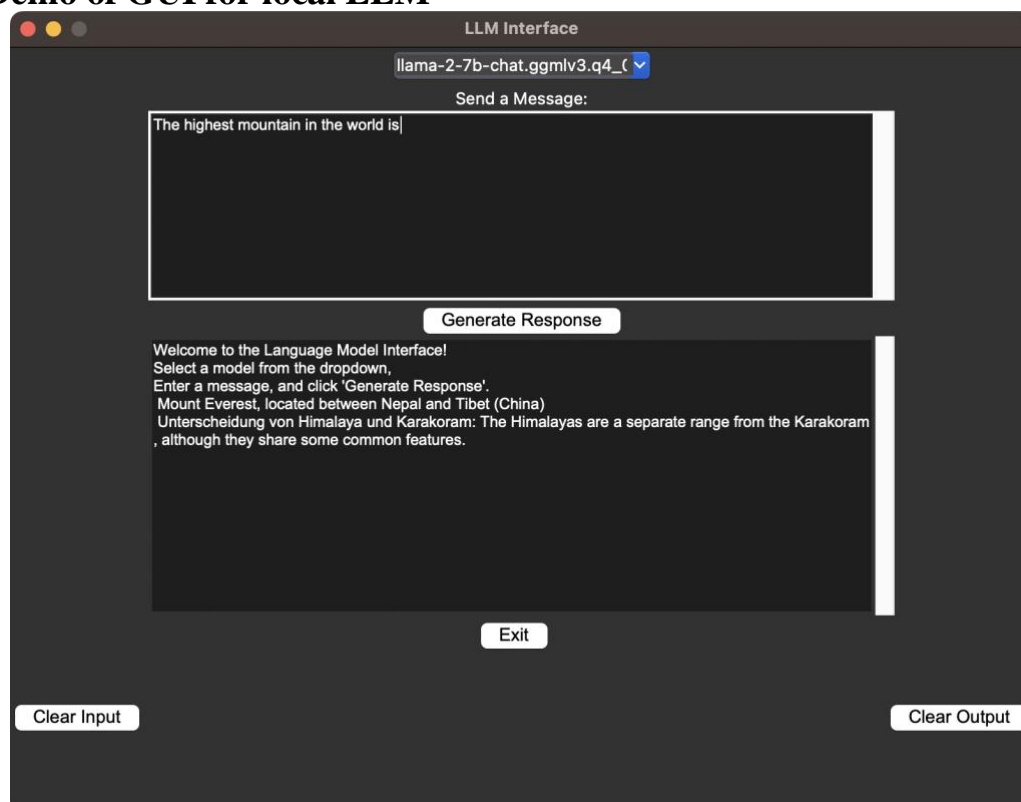
Note: The distances given are for the average distance between Earth and Moon, and Ea
rth and Mars, respectively. These distances can vary slightly due to the elliptical s
hape of their orbits
You: then the distance between earth and sun is
Model: Model: about 149.6 million kilometers.
You: the distance between earth and Mercusys
Model: is Model: about 78 million kilometers.

Note: The distances given are for the average distance between Earth and Sun, and Ear
th and Mercury, respectively. These distances can vary slightly due to the elliptical
shape of their
You: exit
Goodbye!
ggml_metal_free: deallocating

[ Apple ~/W/NU/EE47/Project2_A/LLM 2m 57s Dev_LLM 12:50:46 AM ]
```

## Task 3

### 1. Demo of GUI for local LLM



## Reference

A few useful manuals and documents are listed below for your ease reference, understanding necessary domain of knowledge and finishing the project smoothly.

- [1] The python version <https://www.python.org/downloads/release/python-360/>
- [2] The Pycharm version <https://www.jetbrains.com/pycharm/download/#section=windows>
- [3] The Python Package Index <https://pypi.org/>
- [4] Speech recognition with Python <https://realpython.com/python-speech-recognition/>
- [5] Li, Jiwei, et al. "A diversity-promoting objective function for neural conversation models." arXiv preprint arXiv:1510.03055, 2015.
- [6] Wei, Zhongyu, et al. "Task-oriented dialogue system for automatic diagnosis." Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2018.
- [7] GitHub, <https://github.com/search?q=&type=>
- [8] Anand, Yuvanesh, et al. "Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo." GitHub (2023).
- [9] Touvron, Hugo, et al. "LLaMA: open and efficient foundation language models." arXiv preprint arXiv: 2302.13971 (2023).
- [10] Du, Zhengxiao, et al. "Glm: General language model pretraining with autoregressive blank infilling." arXiv preprint arXiv:2103.10360 (2021).