新加坡国立大学 EE5112：人机交互

项目：开发对话系统及专属大型语言模型（LLM）平台（占总成绩20%）

一、说明

1. 本项目为团队合作任务，旨在体现协作能力，同时允许每位成员自主选择任务模块并明确声明，以展现个人特长。学生需以3人小组形式完成（特殊情况可2人一组）。
2. 将所有文件压缩为一个压缩包，文件命名格式为“project_1_小组编号.zip”。提交至Canvas平台下“Student submission: Project 1 prof Sam Ge”文件夹中。
   示例：若小组编号为1，则文件需命名为“project_1_group_1.zip”。
3. 封面需注明所有成员的姓名及学号。每组仅需提交一份合并报告，建议页数为5-15页，格式要求为：1.5倍行距、12号“Times New Roman”字体、1英寸页边距。压缩包内需包含PDF

格式的报告及Python项目文件，并为所有引用来源添加规范的引用标注。
4. 提交截止日期为2025年9月29日，逾期提交将按学校逾期提交政策处理。
5. 小组成员可自行分配任务，但需确保任务分配均匀，以便充分评估每位成员的贡献。
6. 虽为团队项目且鼓励协作，但每位学生必须在报告对应章节中明确说明个人贡献与付出。报告内容雷同将面临处罚及/或学校纪律处分。
7. 如有任何问题，可联系研究生助教（GA）：
    - 张奥谦（E1144122@u.nus.edu）
    - 张斌杰（E1106673@u.nus.edu）
    - 黄东（E1143962@u.nus.edu）

二、项目目标

对话系统在输入和输出通道中可采用文本、语音、图形、触觉、手势等一种或多种模态进行交互。本手册将提供必要指导，实验助理也将提供有限支持。项目目标如下：

1. 熟悉对话系统的开发流程；
2. 熟悉项目工作环境；
3. 熟悉并安装常用Python库；
4. 熟悉主流开发平台（如TensorFlow）；
5. 熟悉主流开源大型语言模型（如Llama、GLM等）；
6. 开发对话系统及本地LLM平台；
7. 熟悉LLM的评估流程；
8. 积累对话系统开发中的问题发现与解决经验；
9. 注意：每组仅需提交一份完整的对话系统文件，成员贡献需在Python文件中注明。

三、任务内容

通过本章学习，你应已掌握人机对话系统、自然语言处理及大型语言模型的工作原理。首先，以基于Reddit评论数据集的`chatbot.py`文件为示例对话系统，开发符合个人兴趣的实用对话系统；其次，以“GPT4ALL”库为示例，搭建并开发专属本地LLM；最后，为深入理解人机交互（HRI），需开发类ChatGPT交互界面。

任务1：开发符合个人兴趣的对话系统

i. （学生A）根据兴趣查找相关文献或网络资源，学习如何设计涉及自然语言处理的对话系统。
ii. （学生A）根据对话系统主题，查找合适的数据集与模型（或预训练数据集与预训练模型），尝试调整模型参数以获得更优结果。
iii. （团队）开发符合个人兴趣的对话系统，例如：商店、餐厅、诊所接待员等。
iv. （团队）分析对话系统性能，在报告中展示所有结果与发现，并进行讨论与评述。

任务2：基于开源LLM开发本地对话系统

i.（学生B）对不同类型的LLM（编码器-解码器型、仅编码器型、仅解码器型）进行文献综述，总结学习成果并对比不同类型LLM的特点。
ii.（团队）搭建LLM环境并选择模型：安装一个开源库（如gpt4all、llama-cpp-python等），确保可通过命令行与LLM进行交互；尝试不同预训练模型并对比其性能。
iii.（团队）设计支持多轮对话的专属对话系统，核心功能为：按回车键确认输入并等待模型输出，模型响应后可继续输入文本。注意：本任务要求对话可在终端（Terminal）中正常运行。
iv.（学生B）分析对话系统性能，在报告中展示所有结果与发现，并进行讨论与评述。

任务3：LLM性能评估

i.（团队）从GPT4All支持的模型中选择一款LLM，并从下图所示数据集中选择一个，对所选模型与数据集进行评述。

性能基准表

模型	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	平均值
GPT4AH-J6Bv1.0	73.4	74.8	63.4	64.7	54.9	36.0	40.2	58.2
GPT4AE-J v1.1-brezy	74.0	75.1	63.2	63.6	55.4	34.9	38.4	57.8
GPT4All-Jv1.2-jazzy	74.8	74.9	63.6	63.8	56.6	35.3	41.0	58.6
GPT4A1-J v1.3-groovy	73.6	74.3	63.8	63.5	57.7	35.0	38.8	58.1
GPT4AII-J Lora 6B	68.6	75.8	66.2	63.5	56.4	35.7	40.2	58.1
GPT4ALILaMaLora7B	73.1	77.6	72.1	67.8	51.1	40.4	40.2	60.3
GPT4AII 13B snoozy	83.3	79.2	75.0	71.3	60.9	44.2	43.4	65.3
GPT4All Falcon	77.6	79.8	74.9	70.1	67.9	43.4	42.6	65.2
Nous-Hermes	79.5	78.9	80.0	71.9	74.2	50.9	46.4	68.8
Nous-Hermes2	83.9	80.7	80.1	71.3	75.7	52.1	48.2	70.0
Nous-Purfin	81.5	80.7	80.4	72.5	77.6	50.7	45.6	69.9
Dolly68	68.8	77.3	67.6	63.9	62.9	38.7	41.2	60.1
Dolly 12B	56.7	75.4	71.0	62.2	64.6	38.5	40.4	58.4

ii. （团队）下载所选数据集，使用学号后3位作为随机种子，从验证集中随机抽取500个样本，计算所选样本的准确率并写入报告。将数据集处理及评估流程的代码附加至最终压缩包中。也可选择更多模型与数据集，对比不同模型在不同数据集上的性能。
以下为BoolQ数据集简介：

iii. （团队）记录实验过程中遇到的问题、解决方案及经验总结。

任务4：为本地LLM设计图形用户界面（GUI）

i. （学生C）选择一款Python GUI库（如tkinter），评述其功能特点。
ii. （团队）设计类ChatGPT的交互界面。
iii. （团队）确保界面支持多轮对话（如聊天机器人），并具备调整模型、清空对话框等基础功能。
iv. （学生C）记录实验过程中遇到的问题、解决方案及经验总结。

任务5：探索多模态大型语言模型（MLLMs）

近年来，多模态大型语言模型（如GPT-4o及类似架构）在各类任务中展现出卓越性能，下游任务包括融合文本、语音与视觉的交叉对话、理解上下文的复杂图像生成、图像问答（IQA）等。多模态大型语言模型的出现标志着人工智能领域的重大突破，其通过融合多模态信息，提供更具交互性与上下文感知能力的输出。

本任务要求学生探索并理解多模态大型语言模型的基础原理，深入研究其架构、训练流程及实际应用。LLaVA（大型语言与视觉助手）可作为本探索任务的实例，据其项目页面（https://llava-vl.github.io）介绍，LLaVA展示了多模态系统如何通过微调增强视觉-语言理解能力，尤其在图像描述、视觉对话、问答等领域效果显著。

i. （个人）采用文字与图表结合的方式，阐述多模态大型语言模型（MLLMs）与传统LLM的差异，每位学生的回答需控制在1页内。
ii. （团队）从Model Zoo（https://github.com/haotian-liu/LLaVA/blob/main/docs/MODEL_ZOO.md）中选择一款预训练模型，展示其在至少两项任务（如图文对话、图像描述）中的性能实例。

四、分数分布

本项目将从**创新性**、**实用性**、**团队协作**三个核心维度对学生进行全面评估：创新性评估学生解决问题的创意与新思路；实用性评估解决方案在实际场景中的应用价值；团队协作评估成员间的协作效率、沟通质量及贡献度。以上维度将作为综合评价学生项目贡献的重要依据。

不同人数小组的任务分数分布如下：

3人小组

任务	分数占比	负责成员
1	20%	学生A及全体成员
2	20%	学生B及全体成员
3	20%	全体成员
4	20%	学生C及全体成员
5	20%	全体成员

2人小组

任务	分数占比	负责成员
1	20%	学生A及全体成员
2	20%	学生B及全体成员
3	20%	全体成员
4	20%	全体成员
5	20%	全体成员

五、对话系统参考方向

1. 商店、餐厅、诊所等场景的接待员；
2. 数学、物理、编程等学科的辅导助手；
3. 青年医疗咨询助手；
4. 机器人学、人工智能、人机交互等领域的青年教授模拟助手；
5. 商业咨询助手；
6. 青年律师咨询助手；
7. 其他（可自主拓展）。

六、指导示例

任务1

1. 熟悉对话系统开发流程
   如图1所示，涉及自然语言处理的人机对话系统典型活动周期包括以下阶段：
   i. 用户输入（如语音），系统输入识别/解码器（含自动语音识别、手势识别、手写识别等）将其转换为纯文本；
   ii. 自然语言理解（NLU）单元对文本进行分析，包括实体识别、词性标注、句法/语义解析等；
   iii. 对话管理器分析语义信息，记录对话历史与状态，管理对话整体流程；
   iv. 输出渲染器生成输出内容，包括文本转语音引擎、虚拟头像、机器人等形式。

   （注：原文此处附图1“人机对话系统设计流程”，含个性化偏好、问答系统、代理服务、网络服务、知识服务器、互联网、对话历史、对话模型语义、语音输出、文本转语音合成（TTS）、自然语言生成（NLG）、对话管理器（DM）、自然语言输出、语音输入、自动语音识别（ASR）、自动语音理解（NLU）、用户建模模块、文本输入、声学模型、语言模型、语义模型、更新后的用户画像等元素）

2. 配置工作环境

   - 在macOS上安装Anaconda
     从以下链接下载Anaconda图形化安装包：
     （注：原文此处含Anaconda下载页面描述，包括“免费下载”“数据科学基础工具包”“支持多平台部署”等信息）
     安装包版本：Anaconda3 2023.03-1
     安装步骤：选择“仅为我安装”，安装路径为用户主文件夹（需3.55GB存储空间，仅当前用户可使用）。
     详细安装说明参见此链接。

   - 在Windows上安装Anaconda
     从以下链接下载Anaconda图形化安装包：
     （注：原文此处含Windows版本Anaconda下载页面描述，支持Python 3.11）
     详细安装说明参见此链接。

   - 安装VSCode编辑器
     下载链接：[此处附下载链接]
     （注：原文此处含VSCode功能描述，如智能感知、运行调试、内置Git、扩展插件等）

   - 创建Python环境

     1. 在终端验证Anaconda安装：输入`conda -V`，预期输出如`conda 23.5.2`；
     2. 查看现有环境：输入`conda info --envs`，预期输出如`base /Users/zhaohhy/miniconda3`；
     3. 创建项目专属环境：输入`conda create -n your_env_name python=3.9`（示例：`conda create -n project2 python=3.9`）；
     4. 出现安装确认提示时输入`y`；

5. 验证新环境：输入`conda info --envs`，预期新增`project2 /Users/zhaohhy/miniconda3/envs/project2`；
6. 激活环境：输入`conda activate project2`。

3. 熟悉并安装所需Python库

- `tensorflow==2.13`：开源高性能数值计算库，支持跨平台（CPU、GPU、TPU）部署，适用于桌面端、服务器集群、移动设备等场景。
- `numpy`：Python数组计算基础库。
- `gTTS`（Google Text-to-Speech）：调用谷歌翻译文本转语音API的Python库与命令行工具，可生成MP3音频文件或字节流。
- `PyAudio`：PortAudio跨平台音频I/O库的Python绑定，支持在Linux、Windows、macOS等系统上播放与录制音频。
- `SpeechRecognition`：支持多种在线/离线引擎与API的语音识别库。
- TensorFlow安装命令：
  `pip install tensorflow-cpu==2.13.0`
  （注：若需训练自定义模型，需安装GPU版本TensorFlow及兼容CUDA的高性能GPU）
- PyAudio安装（macOS）：
  1. 用Homebrew安装PortAudio：`brew install portaudio`；
  2. 安装PyAudio：`pip install pyaudio`；
     注意事项：需提前安装Xcode命令行工具，pip将自动编译PyAudio源码。
     安装成功验证：[此处附验证方法]
- SpeechRecognition安装命令：`pip install SpeechRecognition`

4. 开始实践
   步骤1：运行预训练模型
   下载预训练模型：https://drive.google.com/uc?id=1rRRY-y1KdVk4UB5qhu7BjQHtfadIOmMk&export=download
   解压后得到“reddit”文件夹，将其放入项目的“models”目录中。
   项目文件结构如下：

```
├── _pycache_
├── data
├── LICENSE.txt
├── model.py
├── models
│   ├── folder-readme.txt
│   └── reddit
├── README.md
├── reddit-parse
```

```
├── chatbot.py
├── train.py
└── utils.py
```

步骤2：运行`chatbot.py`文件

```
~/Documents/EE5112_Project2>python chatbot.py
2023-09-12 16:27:38.550431: I tensorflow/core/platform/cpu_feature_guard.cc:182] This is
deprecated and will be removed in a future version. Instructions for updating: non-
resource variables are not supported in the long term
Creating model... /Users/zhaohhy/miniconda3/envs/project2/lib/python3.9/site-
packages/tensorflow/python/... (省略部分警告信息)
```

调整`chatbot.py`参数以优化结果：
a. `beam_width`：默认使用束宽为2的束搜索生成响应，增大束宽可获得更严谨但速度更慢的响应，设为1可禁用束搜索；
b. `temperature`：调节字符概率分布，1.0为默认中性值，值越小结果越保守，值越大越灵活，建议范围0.5-1.5（超出范围可能导致结果不连贯）；
c. `top-n`：仅保留前n个高概率字符，默认禁用；
d. `relevance`：并行运行主模型与掩码模型，掩码模型权重由该参数控制，按文献公式（Li, Jiwei等人，2015）融合概率，旨在提升对话相关性（避免“我不知道”等通用回复）。值越高相关性越强但响应连贯性可能下降，建议不超过0.4，负值禁用（默认），启用后速度减半。

聊天过程中可实时调整参数（无需重启程序）：

- 设置温度：`--temperature 1.3`（输出：[Temperature set to 1.3]）
- 设置相关性：`--relevance 0.3`（输出：[Relevance set to 0.3]）
- 禁用相关性：`--relevance -1`（输出：[Relevance disabled]）
- 禁用top-n过滤：`--topn -1`（输出：[Top-n filtering disabled]）
- 设置top-n=2：`--topn 2`（输出：[Top-n filtering set to 2]）
- 重置模型状态：`--reset`（输出：[Model state reset]）

对话示例（非筛选结果，“>”为用户输入，无“>”为系统响应）：

> Hi, how are you?
> I'm still an adventurer like you. I just didn't want you to think it was real. How old are you? How old are you? Could you give me a joke?
> Think of all the conversations you have to say when you think you're an adult.
> Well, you're the one that doesn't know what you're talking about, do you? Who built you?
> Oh, I don't know what that means. no
> Okay fine. Thank you for your patience. I really appreciate it.
> Could you write a poem for me?

> Could you tell me what you mean? I guess you cannot do that.
> Yeah I don't know what you're talking about.

任务2

以下为两款开源LLM库，可任选其一使用：

1. GPT4All入门
   免费本地运行的隐私保护型聊天机器人，无需GPU或网络。
   官网链接：https://gpt4all.io/index.html（提供Windows、macOS、Ubuntu安装包）
   安装流程：[此处附安装向导截图描述，含“准备解压组件”“安装”等步骤]
   Python安装命令：`pip install gpt4all`
   项目部署参考：https://github.com/nomic-ai/gpt4all/tree/main/gpt4all-bindings/python

   安装示例：

   ```
   ~/Documents/EE5112_Project2>pip install gpt4all
   Collecting gpt4all
     Downloading gpt4all-1.0.10-py3-none-macosx_10_9_universal2.whl (6.0 MB)
     Obtaining dependency information for gpt4all from https://files.pythonhosted.org/...
   Requirement already satisfied: requests in
   /Users/zhaohhy/miniconda3/envs/project2/lib/python3.9/site-packages (from gpt4all)
   Collecting tqdm (from gpt4all)
     Downloading tqdm-4.66.1-py3-none-any.whl (78 kB)
   Installing collected packages: tqdm, gpt4all
   Successfully installed gpt4all-1.0.10 tqdm-4.66.1
   ```

   简单示例代码：

   ```python
   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)  # 预期输出：Paris
   ```

   （预训练模型将自动下载至本地）

2. llama-cpp-python入门
   LLaMA（Meta AI大型语言模型）是尖端基础模型，旨在助力研究者推进AI领域研究。其轻量化、高性能版本降低了基础设施门槛，促进该领域的民主化发展。
   llama-cpp-python：适用于CPU环境的开源库。
   安装命令：`pip install llama-cpp-python`
   详细说明参见：[此处附链接]

   安装与使用示例：

```
# 安装命令
pip install llama-cpp-python

# 高阶API使用
from llama_cpp import Llama
llm = Llama(model_path="./models/7B/ggml-model.bin")
output = llm("Q: Name the planets in the solar system? A:", max_tokens=32, stop=["Q:",
"\n"], echo=True)
print(output)
# 预期输出：
# {
#   "id": "cmpl-xxxxxx",
#   "object": "text_completion",
#   "created": 1679561337,
#   "model": "./models/7B/ggml-model.bin",
#   "choices": [
#     {
#       "text": "Q: Name the planets in the solar system? A: Mercury, Venus, Earth,
Mars, Jupiter, Saturn, Uranus, Neptune",
#       "index": 0,
#       "logprobs": None,
#       "finish_reason": "stop"
#     }
#   ],
#   "usage": {
#     "prompt_tokens": 14,
#     "completion_tokens": 28,
#     "total_tokens": 42
#   }
# }
```

3. LLM多轮对话演示

   可与模型启动对话，按回车键确认输入，输入“exit”结束对话：

```
You: the distance between earth and moon is
Model: about 384,000 kilometers. Note: The distance between Earth and Moon varies
slightly due to the elliptical shape of their orbits around each other. At its closest
point (called perigee), the Moon is about 363,000 kilometers from Earth, and at its
farthest point (apogee), it is about 406,000 kilometers away.
You: the distance between earth and Mars
Model: The average distance between Earth and Mars is about 225 million kilometers.
Note: The distances given are for the average distance between Earth and Moon, and Earth
and Mars, respectively. These distances can vary slightly due to the elliptical shape of
their orbits.
You: the distance between earth and Mercury
Model: The average distance between Earth and Mercury is about 78 million kilometers.
Note: The distances given are for the average distance between Earth and Sun, and Earth
and Mercury, respectively. These distances can vary slightly due to the elliptical shape
of their orbits.
You: exit
Goodbye!
ggml_metal_free: deallocating
```

任务3 LLM模型评估

以下为GPT4All技术报告中的性能表，可选择一款基础模型及一个数据集进行评估。

表1：常识推理任务零样本性能（每列中性能最优的非OpenAI模型已加粗）

模型	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	平均值
GPT4All-J 6B v1.0	73.4	74.8	63.4	64.7	54.9	36.0	40.2	58.2
GPT4All-J v1.1-breezy	74.0	75.1	63.2	63.6	55.4	34.9	38.4	57.8
GPT4All-J v1.2-jazzy	74.8	74.9	63.6	63.8	56.6	35.3	41.0	58.6
GPT4All-J v1.3-groovy	73.6	74.3	63.8	63.5	57.7	35.0	38.8	58.1
GPT4All-J Lora 6B	68.6	75.8	66.2	63.5	56.4	35.7	40.2	58.1
GPT4All LLaMa Lora7B	73.1	77.6	72.1	67.8	51.1	40.4	40.2	60.3
GPT4All 13B spoozy	83.3	79.2	75.0	71.3	60.9	44.2	43.4	65.3
Dolly 6B	68.8	77.3	67.6	63.9	62.9	38.7	41.2	60.1
Dolly 12B	56.7	75.4	71.0	62.2	64.6	38.5	40.4	58.4
Alpaca 7B	73.9	77.2	73.9	66.1	59.8	43.3	43.4	62.4
Alpaca Lora 7B	74.3	79.3	74.0	68.8	56.6	43.9	42.6	62.8
GPT-J 6.7B	65.4	76.2	66.2	64.1	62.2	36.6	38.2	58.4
LLaMA 7B	73.1	77.4	73.0	66.9	52.5	41.4	42.4	61.0
LLaMA 13B	68.5	79.1	76.2	70.1	60.0	44.6	42.2	63.0
Pythia 6.7B	63.5	76.3	64.0	61.1	61.3	35.2	37.2	57.0
Pythia 12B	67.7	76.6	67.3	63.8	63.9	34.8	38.0	58.9
Fastchat TS	81.5	64.6	46.3	61.8	49.3	33.3	39.4	53.7
Fastchat Vicuña 7B	76.6	77.2	70.7	67.3	53.5	41.2	40.8	61.0

模型	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	平均值
Fastchat Vicuña 13B	81.5	76.8	73.3	66.7	57.4	42.7	43.6	63.1
Stable Vicuña RLHF	82.3	78.6	74.1	70.9	61.0	43.5	44.4	65.0
StableLM Tuned	62.5	71.2	53.6	54.8	52.4	31.1	33.4	51.3
StableLM Base	60.1	67.4	41.2	50.1	44.9	27.0	32.0	42.2
Kerala 13B	76.5	77.9	72.6	68.8	54.3	41.0	42.8	62.0
Open Assistant Pythia12B	67.9	78.0	68.1	65.0	64.2	40.4	43.2	61.0
text-davinci-003	88.1	83.8	83.4	75.8	83.9	63.9	51.0	75.7

（注：text-davinci-003在所有任务中仍优于其他模型）

以BoolQ数据集为例，可通过以下链接查看：
https://huggingface.co/datasets/boolq/viewer/default/validation
终端下载命令：`git clone https://huggingface.co/datasets/boolq`

任务4

本任务需编写代码构建对话系统GUI演示，推荐使用Tkinter库。

本地LLM GUI演示

（注：原文此处含GUI界面截图描述，含“LLM Interface”标题、模型选择框（如“llama-2-7b-chat.ggmlv3.q4_”）、消息输入框、“Generate Response”“Exit”“Clear Input”“Clear Output”按钮等元素）

参考链接：

- https://www.tutorialspoint.com/python/python_gui_programming.htm
- https://docs.python.org/3/library/tkinter.html

七、参考资料

以下为实用手册与文档，助力理解相关领域知识及顺利完成项目： [1] Python版本下载：https://www.python.org/downloads/release/python-360/
[2] PyCharm版本下载：https://www.jetbrains.com/pycharm/download/#section=windows
[3] Python包索引：https://pypi.org/
[4] Python语音识别：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.（《神经对话模型的多样性促进目标函数》，arXiv预印本，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.（《面向自动诊断的任务导向对话系统》，第56届计算语言学协会年会论文集（第2卷：短文），2018）
[7] GitHub搜索：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).（《GPT4All：基于GPT-3.5-turbo大规模数据蒸馏训练助手型聊天机器人》，GitHub，2023）
[9] Touvron, Hugo, et al. "LLaMA: open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023).（《LLaMA：开源高效基础语言模型》，arXiv预印本，2023）
[10] Du, Zhengxiao, et al. "Glm: General language model pretraining with autoregressive blank infilling." arXiv preprint arXiv:2103.10360 (2021).（《GLM：基于自回归空白填充的通用语言模型预训练》，arXiv预印本，2021）