用 Intel Extension for Transformers 在 Intel 处理器上搭建自己的 RAG chatbot

Intel Extension for Transformers 介绍:
Intel Extension for Transformers 是 Intel 公司技术人员开发的一种对使用 Intel 计算产品(Xeon、Core、Gaudi 系列芯片和 Intel GPU)进行优化，用于加速基于 Transformer 的大模型的工具包。Intel Extension for Transformers 在 Hugging Face Transformers API 的基础上进行了扩展，提供了一系列特性，包括：提供扩展的 Hugging Face transformers API；先进的大模型软件优化技术和模型压缩感知运行时，基于预训练模型，提供更快速、精准的微调训练和更高效的推理能力；NeuralChat 框架，用户可以在几分钟内用开源大模型创建自己的聊天机器人应用，并提供了丰富的插件选择，来增强聊天机器人功能，例如 Knowledge Retrieval，Speech Interaction, Query Caching 和 Security Guardrail，并且可以配置与 OpenAI SDK 相兼容的 RESTful API；最重要的一点是，提升了 Intel CPU 进行大模型推理的性能，可以基于 Intel CPU 低成本、快捷部署生成式 AI 应用。

Intel Extension for Transformers 使用:
接下来简单介绍如何使用 NeuralChat 框架，基于智谱 AI 开源的 chatglm3 模型快速搭建一个问答机器人。
首先，打开 notebook，通过 git 在 modelscope 下怎么 chatglm3-6b 模型和 embedding 模型:
! git clone https://www.modelscope.cn/ZhipuAI/chatglm3-6b.git
! git clone https://www.modelscope.cn/AI-ModelScope/bge-base-zh-v1.5.git
然后通过 neural_chat 提供的结构配置模型的推理 pipeline:

```
from intel_extension_for_transformers.neural_chat import PipelineConfig
from intel_extension_for_transformers.neural_chat import build_chatbot
from intel_extension_for_transformers.neural_chat import plugins
from intel_extension_for_transformers.transformers import RtnConfig
plugins.retrieval.enable=True
plugins.retrieval.args['embedding_model'] = "./bge-base-zh-v1.5"
plugins.retrieval.args["input_path"]="./mental_health.txt"
config = PipelineConfig(model_name_or_path='./chatglm3-6b',
  plugins=plugins,
  optimization_config=RtnConfig(compute_dtype="int8",
weight_dtype="int4_fullrange"))
chatbot = build_chatbot(config)
```

这里我们指定了开启 RAG 插件，并指定在 RAG 中使用的 embedding 模型为 bge-base-zh-v1.5，然后导入知识库文件。这里的 mental_health.txt 是从 hugging face 上下载的一个心理咨询问答的数据集，每一条数据分为两部分：咨询者的提问 context 和根据提问内容回复的 response。
NeuralChat 框架下的 RAG 知识库支持多种格式的文件：txt 纯文本、特殊要求的 jsonl 格式文件和 xlsx 表格文件等等，但是对于结构化的数据有格式要求，例如：Jsonl 格式文件的内容为{“content”:”xxx”, “link”:0}；xlsx 文件的格式如 sample 所示:

Questions	Answers
Who is the CEO of Intel?	Patrick P. Gelsinger

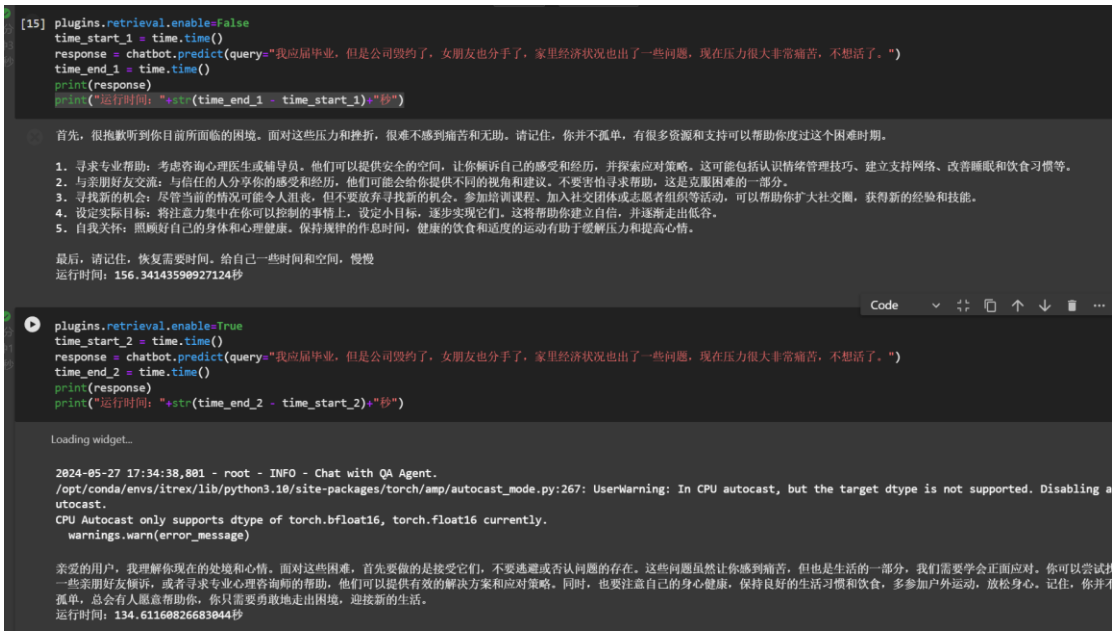
因为单纯的 json 格式导入会不支持，所以我直接写了个脚本把 json 文件的内容转换为 txt

格式存储的问答文本。
optimization_config 是 NeuralChat 的优化选项，这里选择的是设置 weight_dtype 为 int8，通过 weight only quantization 技术优化模型，减少模型中权重和激活值的数值精度，从而减少模型的大小，但保持模型的精度。
配置完成 pipeline 后，通过 build_chatbot(config)就可以初始化一个聊天机器人。

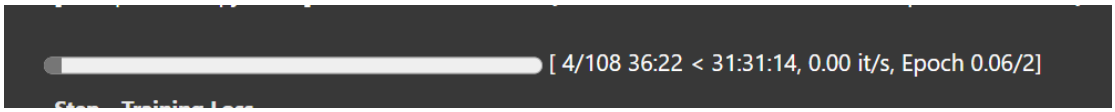
Intel Extension for Transformers 运行过程：
Intel Extension for Transformers 首先实例化插件。对于 RAG 插件，首先将上传的文件进行分词、格式转换和向量化，最后创建 retriever。
接着 Intel Extension for Transformers 会对模型应用 Weight Only Quantization。总共 51 分钟，在 modelscope 提供的免费 cpu 平台上，Intel Extension for Transformers 就会完成 WOQ 的处理，并加载优化后的模型。对于没有显卡资源的用户来讲，我觉得这个时间是可以接受的。

RAG chatbot 效果分析：
通过设置 plugins.retrieval.enable 为 True 或者 False 可以控制 chatbot 实例是否开启或关闭 RAG 插件。通过比较是否开启 RAG，chatbot 生成回答的质量和时间来进行一些简单的分析。

可以很明显看出，大模型直接生成的回答会讲车轱辘话，但是经过检索后生成的回答的精准性就会高很多，会明确指出咨询者的问题所在，提出具体的建议，并对危险行为进行劝阻。启用 RAG 的 chatbot 回答可能还快一点：

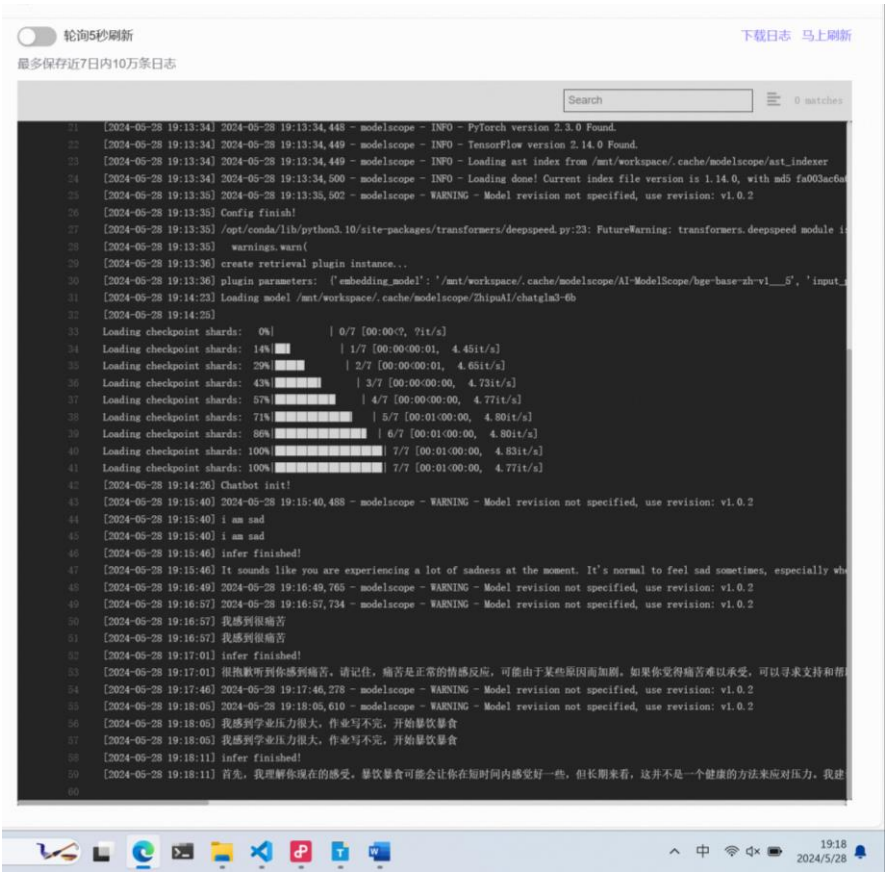


如果 NeuralChat 的 RAG 不能满足要求，可以考虑做 fine tuning。NeuralChat 框架也提供了一个简单的 LoRa 接口，可以用 json 数据集一键微调。仍然使用心理咨询数据集，修改为 alpaca 格式的文件之后，用 NeuralChat 提供的函数对 Yi-1.5-9B-Chat 模型进行微调（Intel Extension for Transformers 暂时不支持 GLM 模型的微调）：

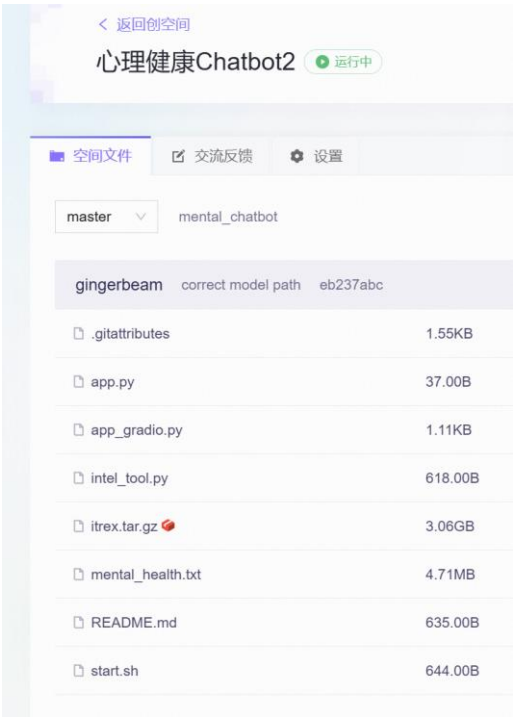


但是在 modelscope 提供的环境中，训练的速度太慢了。

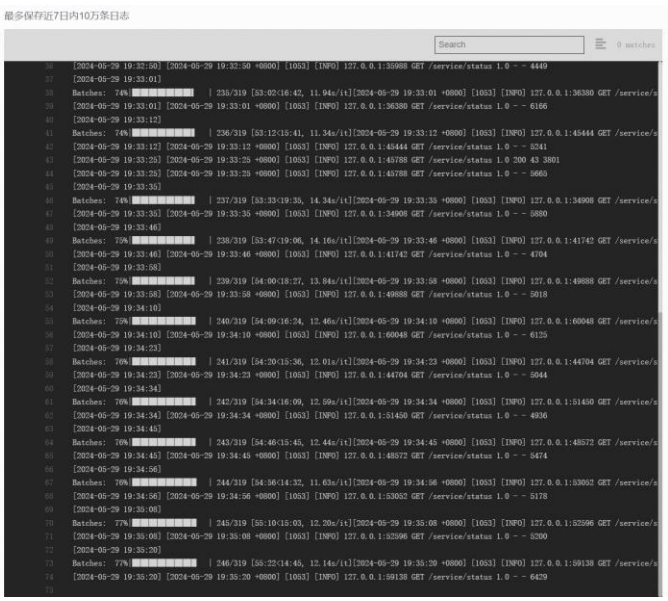
尝试在 ModelScope 上部署 chatbot：
用单 python 文件和 requirements.txt 的形式把文件放到 ModelScope 上面：
心理健康 Chatbot · 创空间（modelscope.cn)
gingerbeam/AskMeAnything2 (github.com)
等待安装完依赖，初始化 chatbot 后，就可以通过 streamlit 与 NeuralChat 的 python 接口交互：

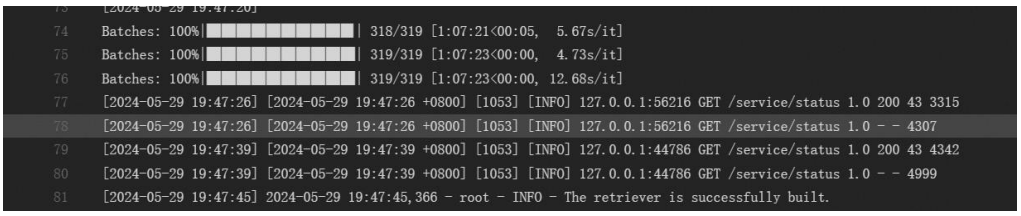


这种方式部署需要下载很多依赖，存在互相不兼容的现象，并且在 ModelScope 平台的受限资源上容易出错，所以也采取了第二种方法：将环境压缩包（wget https://idz-ai.oss-cn-hangzhou.aliyuncs.com/LLM/itrex.tar.gz）作为 lfs 打包进 git 仓库，项目启动后通过 python 启动一个命令行脚本，解压压缩包，激活环境，在部署好的环境中启动 gradio：



这个新的项目部署在[心理健康 Chatbot2 · 创空间 (modelscope.cn)]
加载 RAG 插件：

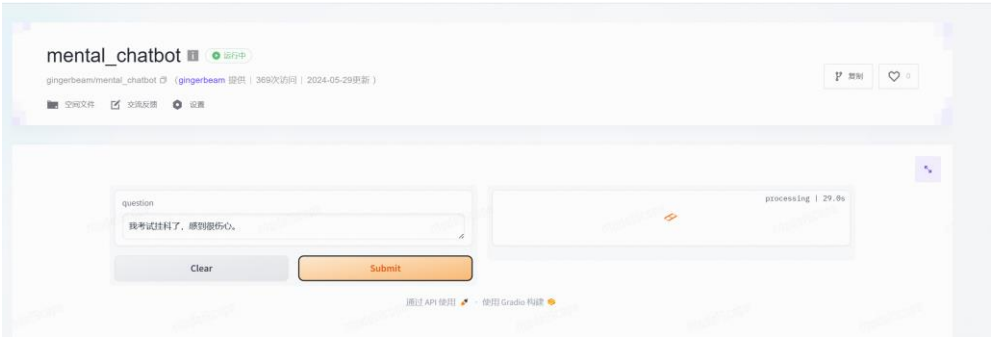




Chatbot 搭建成功：

```
[2024-05-29 19:49:38] [2024-05-29 19:49:38 +0800] [1053] [INFO] 127.0.0.1:35422 GET /service/status 1.0 200 43 3543
[2024-05-29 19:49:38] [2024-05-29 19:49:38 +0800] [1053] [INFO] 127.0.0.1:35422 GET /service/status 1.0 - - 4667
[2024-05-29 19:49:43] 2024-05-29 19:49:43,189 - root - INFO - Model loaded.
[2024-05-29 19:49:43] Chatbot built in app!
[2024-05-29 19:49:43] 2024-05-29 19:49:43,539 - httpx - INFO - HTTP Request: GET https://checkip.amazonaws.com/ "HTTP/1.1 200 "
[2024-05-29 19:49:44] 2024-05-29 19:49:44,147 - httpx - INFO - HTTP Request: GET https://api.gradio.app/pkg-version "HTTP/1.1 200 OK"
[2024-05-29 19:49:44] Running on local URL:  http://127.0.0.1:7860
[2024-05-29 19:49:44] 2024-05-29 19:49:44,265 - httpx - INFO - HTTP Request: GET http://127.0.0.1:7860/startup-events "HTTP/1.1 200 OK"
[2024-05-29 19:49:44] 2024-05-29 19:49:44,288 - httpx - INFO - HTTP Request: HEAD http://127.0.0.1:7860/ "HTTP/1.1 200 OK"
[2024-05-29 19:49:45] 2024-05-29 19:49:45,221 - httpx - INFO - HTTP Request: GET https://api.gradio.app/v2/tunnel-request "HTTP/1.1 200 OK
[2024-05-29 19:49:48] [2024-05-29 19:49:48,641][INFO] change status to RUNNING
[2024-05-29 19:49:50] [2024-05-29 19:49:50 +0800] [1053] [INFO] 127.0.0.1:51776 GET /service/status 1.0 200 42 4707
[2024-05-29 19:49:50] [2024-05-29 19:49:50 +0800] [1053] [INFO] 127.0.0.1:51776 GET /service/status 1.0 - - 5297
[2024-05-29 19:50:15]
[2024-05-29 19:50:15] Could not create share link. Missing file: /opt/conda/envs/itrex/lib/python3.10/site-packages/gradio/frpc_linux_amd6
[2024-05-29 19:50:15]
[2024-05-29 19:50:15] Please check your internet connection. This can happen if your antivirus software blocks the download of this file.
[2024-05-29 19:50:15]
[2024-05-29 19:50:15] 1. Download this file: https://cdn-media.huggingface.co/frpc-gradio-0.2/frpc_linux_amd64
[2024-05-29 19:50:15] 2. Rename the downloaded file to: frpc_linux_amd64_v0.2
[2024-05-29 19:50:15] 3. Move the file to this location: /opt/conda/envs/itrex/lib/python3.10/site-packages/gradio
```

在 Gradio 搭建的界面上进行交互：




```
[2024-05-29 19:50:55] 2024-05-29 19:50:55,865 - matplotlib.font_manager - INFO - generated new fo
[2024-05-29 19:53:11]
Batches:   0%|          | 0/1 [00:00<?, ?it/s]
Batches: 100%|██████████| 1/1 [00:00<00:00,  7.02it/s]
Batches: 100%|██████████| 1/1 [00:00<00:00,  7.00it/s]
[2024-05-29 19:53:11] 2024-05-29 19:53:11,826 - root - INFO - Chat with QA Agent.
```

经过推理后生成了回答：

```
[2024-05-29 19:50:55] 2024-05-29 19:50:55,697 - matplotlib.font_manager - WARNING - Matplotlib is building the font cache; this may take a
[2024-05-29 19:50:55] 2024-05-29 19:50:55,865 - matplotlib.font_manager - INFO - generated new fontManager
[2024-05-29 19:53:11]
Batches:   0%|          | 0/1 [00:00<?, ?it/s]
Batches: 100%|██████████| 1/1 [00:00<00:00,  7.02it/s]
Batches: 100%|██████████| 1/1 [00:00<00:00,  7.00it/s]
[2024-05-29 19:53:11] 2024-05-29 19:53:11,826 - root - INFO - Chat with QA Agent.
[2024-05-29 20:00:32] Based on your symptoms and the information available, it seems like you may be experiencing some form of mental heal
```

上线的项目还存在两个问题：

一是语言问题，这可能因为采用的数据集是英文数据集，也可能是 RAG 无法准确理解“挂科”推理 pipeline 配置的过程可能有一些问题；还有一个是 pipeline 没有配置模型优化，这主要

是因为免费云环境只有 16GB 内存，担心会爆内存。

Anyway，这是一个使用 Intel Extension for Transformers 轻松搭建聊天机器人应用的成功实践，可以看到 Intel Extension for Transformers 在 Intel CPU 上起到的加速效果和工程实践上的便利之处。这个项目未来可以调整的地方还有很多，例如数据集的调整和进一步发掘 Intel Extension for Transformers 的能力，在 Intel 设备上更简单高效地搭建个人大模型定制服务。