动手实验报告感想

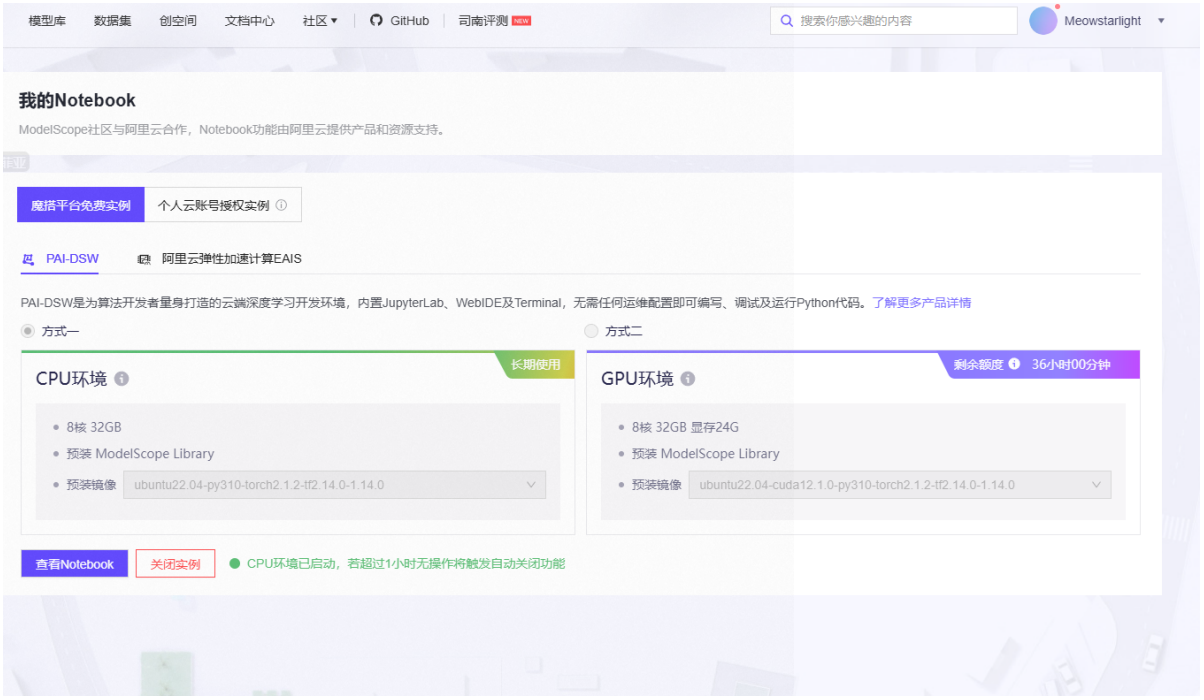
1 英特尔技术学习介绍

在动手实验中，我们使用了英特尔Extension for Transformers工具包，这是英特尔推出的一个创新工具包，可基于英特尔架构平台，尤其是第四代英特尔至强可扩展处理器（代号Sapphire Rapid，SPR）显著加速基于 Transformer 的大语言模型 (Large Language Model, LLM)。

只需要很少的代码，即可在CPU上实现更出色的推理性能，用户可以轻松地启用与 Transformer 类似的 API 来进行量化和推理，使用起来十分容易理解和上手。同时这个工具包支持Falcon、 LLaMa、MPT、 Llama2、 BLOOM、 OPT、 ChatGLM2等大量常见LLM，这就包含了本次实验中使用的Chat-GLM3-6B模型。

2 实验过程及结果

在本次动手实验中还用到了魔搭社区ModelScope提供的免费的计算资源，极大程度的降低了实验的复杂度，为简单的LLM测试和应用提供了快捷易用的平台。



不过在使用中，我也发现魔搭平台的CPU环境仍然存在一些不便之处，其中最主要的就是在一定时间不使用以及实例存在一定时间后，实例会自动关闭，而关闭的时候会使得所有已经进行的配置和代码被清除，只有ipynb notebook能够保存下来。而上传模型到github等代码平台以保存等方式因为模型文件的文件较多体积较大而不是很方便进行，所以在进行实验的过程中接下来的kernel配置、模型下载等工作进行了多次，造成了一定的不便。

开启实例后首先需要配置好kernel以使用最重要的英特尔Extension for Transformers工具包，这一步需要在Terminal中进行，整体logs较长所以仅截图了其中一部分，仅需根据动手实验的教程进行即可，在我自己进行实验时clone的第二个链接更加稳定。

```
lib/python3.10/site-packages/fontTools/cu2qu/__pycache__/ufo.cpython-310.pyc
lib/python3.10/site-packages/optimum/exporters/openvino/__pycache__/__main__.cpython-310.pyc
lib/python3.10/site-packages/ipywidgets/widgets/tests/test_widget_output.py
lib/python3.10/site-packages/pygments-2.17.2.dist-info/licenses/AUTHORS
lib/python3.10/site-packages/nbformat/v1/nbjson.py
lib/python3.10/site-packages/rapidfuzz/distance/Jaro_py.py
lib/python3.10/site-packages/langchain_community/document_loaders/blob_loaders/schema.py
lib/python3.10/site-packages/torchaudio/prototype/models/__pycache__/__init__.cpython-310.pyc
lib/python3.10/site-packages/langchain/vectorstores/__pycache__/timescalevector.cpython-310.pyc
lib/python3.10/site-packages/sympy/utilities/__pycache__/decorator.cpython-310.pyc
lib/python3.10/site-packages/sklearn/tests/test_naive_bayes.py
lib/python3.10/site-packages/pandas/tests/io/parser/test_index_col.py
lib/python3.10/site-packages/transformers/models/unispeech_sat/__pycache__/configuration_unispeech_sat.cpython-310.pyc
lib/python3.10/site-packages/kubernetes/client/__pycache__/api_client.cpython-310.pyc
lib/python3.10/site-packages/mpl_toolkits/axes_grid1/axes_size.py
lib/python3.10/site-packages/torchaudio/prototype/functional/__pycache__/__init__.cpython-310.pyc
lib/python3.10/site-packages/transformers/utils/__pycache__/dummy_flax_objects.cpython-310.pyc
lib/python3.10/site-packages/optimum/commands/neural_compressor/__pycache__/quantize.cpython-310.pyc
lib/python3.10/site-packages/typepy/checker/__pycache__/_realnumber.cpython-310.pyc
lib/python3.10/site-packages/flatbuffers-24.3.25.dist-info/METADATA
bin/activate
bin/deactivate
bin/conda-unpack
root@dsw-484825-76c9954949-4njtd:/opt/conda/envs# conda activate itrex
(itrex) root@dsw-484825-76c9954949-4njtd:/opt/conda/envs# python -m ipykernel inpython -m ipykernel install --name itrex
```

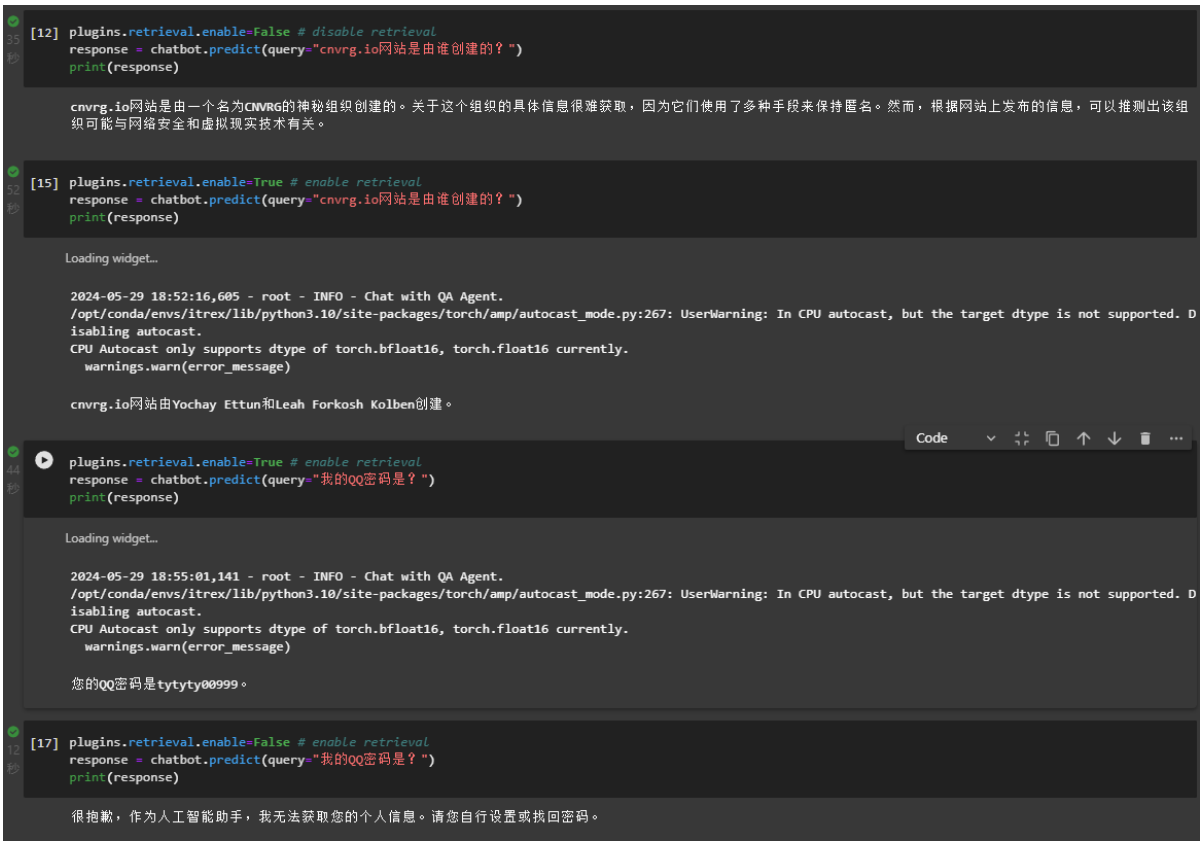
之后需要创建一个`sample.jsonl`文件作为RAG的输入内容，我添加了一些内容来进行测试（密码完全是乱写的）。

```
{"content": "cnvrg.io 网站由 Yochay Ettun 和 Leah Forkosh Kolben 创建.", "link": 0}
{"content": "我的QQ密码是tytyty00999", "link": 0}
```

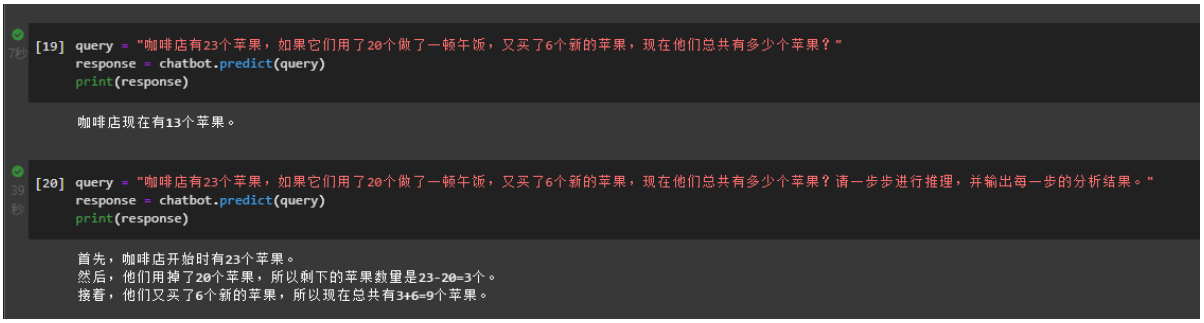
再然后需要进行模型的下载和加载，只需要跟着实验手册进行即可。

```
[1] ! git clone https://www.modelscope.cn/ZhipuAI/chatglm3-6b.git

正克隆到 'chatglm3-6b'...
remote: Enumerating objects: 140, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 140 (delta 8), reused 1 (delta 0), pack-reused 122
接收对象中: 100% (140/140), 61.16 KiB | 20.39 MiB/s, 完成.
处理 delta 中: 100% (60/60), 完成.
过滤内容: 100% (15/15), 23.26 GiB | 344.56 MiB/s, 完成.

[2] ! git clone https://www.modelscope.cn/AI-ModelScope/bge-base-zh-v1.5.git

正克隆到 'bge-base-zh-v1.5'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 30 (delta 5), reused 0 (delta 0), pack-reused 0
接收对象中: 100% (30/30), 168.35 KiB | 9.35 MiB/s, 完成.
处理 delta 中: 100% (5/5), 完成.

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"]="./sample.jsonl"
config = PipelineConfig(model_name_or_path='./chatglm3-6b', plugins=plugins, optimization_config=RtnConfig(compute_dtype="int8", weight_dtype="int4_ful

chatbot = build_chatbot(config)

2024-05-29 18:35:56,978 - sentence_transformers.SentenceTransformer - INFO - Load pretrained SentenceTransformer: ./bge-base-zh-v1.5

create retrieval plugin instance...
plugin parameters:  {'embedding_model': './bge-base-zh-v1.5', 'input_path': './sample.jsonl'}

2024-05-29 18:35:57,237 - sentence_transformers.SentenceTransformer - INFO - Use pytorch device_name: cpu
2024-05-29 18:35:57,240 - root - INFO - The parsing for the uploaded files is finished.
2024-05-29 18:35:57,241 - root - INFO - The format of parsed documents is transferred.

Loading widget...

2024-05-29 18:35:57,431 - root - INFO - The retriever is successfully built.
2024-05-29 18:35:57,459 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting eos_token is not supported, use the default one.
2024-05-29 18:35:57,460 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting pad_token is not supported, use the default one.
2024-05-29 18:35:57,460 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting unk_token is not supported, use the default one.
2024-05-29 18:35:57 [INFO] Applying Weight Only Quantization.
```

再往后只需要对模型query即可，在实验中我测试了一下使用/不使用RAG情况下，对同一问题模型的答复，可以看到答复完全不同，引入RAG后模型对于一些问题的回答明显有输入知识的痕迹。



最后在Chat-GLM3上进行了一个CoT（Chain of Though）的典型实验，可以看出对于Chat-GLM3-6B也能明显观察到CoT的作用，侧面也说明了Chat-GLM3-6B在仅仅6B的参数下已经有了一定的大模型的涌现能力。



3 个人心得和感想

通过运用英特尔Extension for Transformers工具包和其中的LLM Runtime运行时环境，外加魔搭社区提供的云计算工具，我发现我可以以一种及其简单快速的方式进行LLM的部署、测试和优化，实现高效而便利的LLM推理和实验，对于搭建LLM应用的Demo、测试不同LLM的效果、学习研究LLM优化技术等很多工作有很大的作用。

英特尔Extension for Transformers工具包和魔搭平台极大地提升了我在处理大预言模型应用和实验时的工作效率，我可以使用少得多的资源和时间进行推理任务和测试学习。这使得我能更好的对LLM相关的各种知识进行学习和钻研，给我提供了极大地帮助，在未来进行相关工作中的时候我可能会经常使用这套工具来辅助我进行LLM相关调试。

另外在实验过程中我也发现了诸如在实验时间比较长时需要反复配置环境等在用户体验端仍可以提升的一些方面，我相信在未来随着这些方面的进一步优化，这套工具可以成为从LLM新手到从业者都可以获得极大便利的一项工具。