

大语言模型部署报告

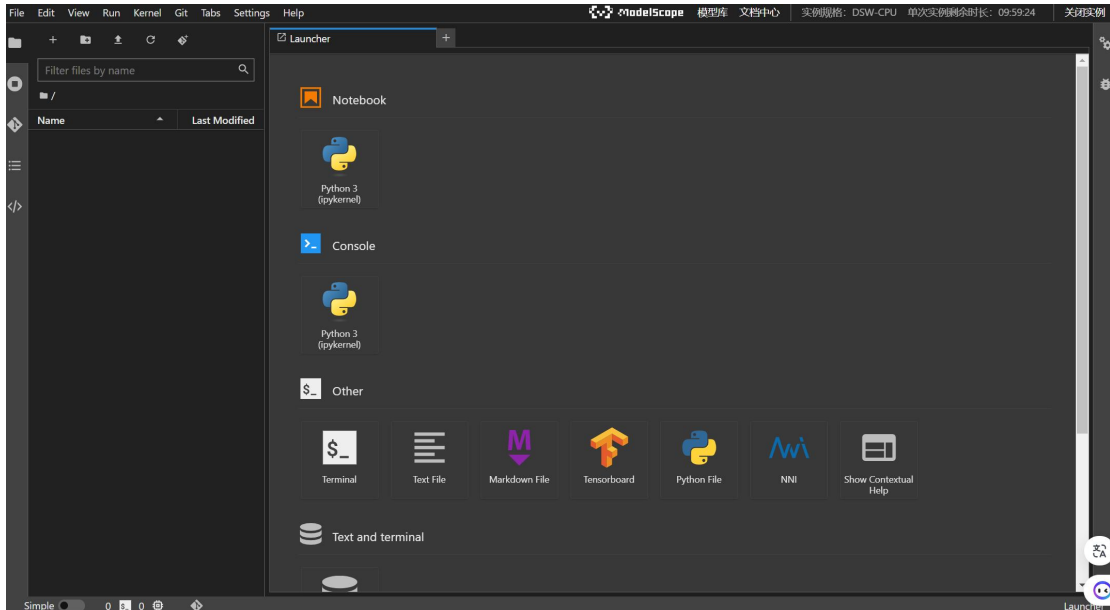
姓名：林觉凯

学号：2253744

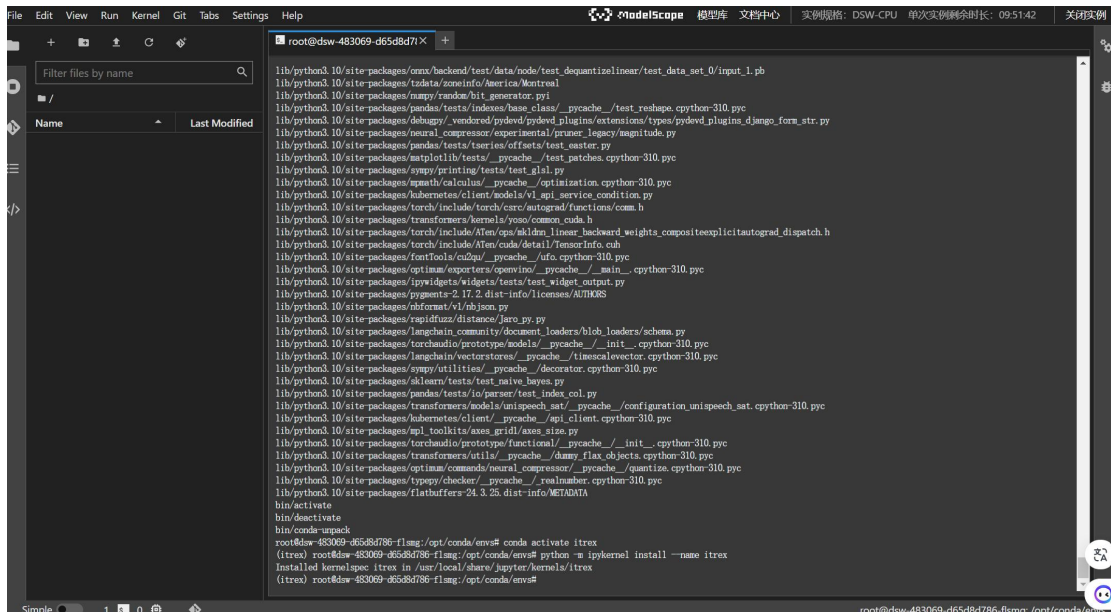
指导老师：曾进

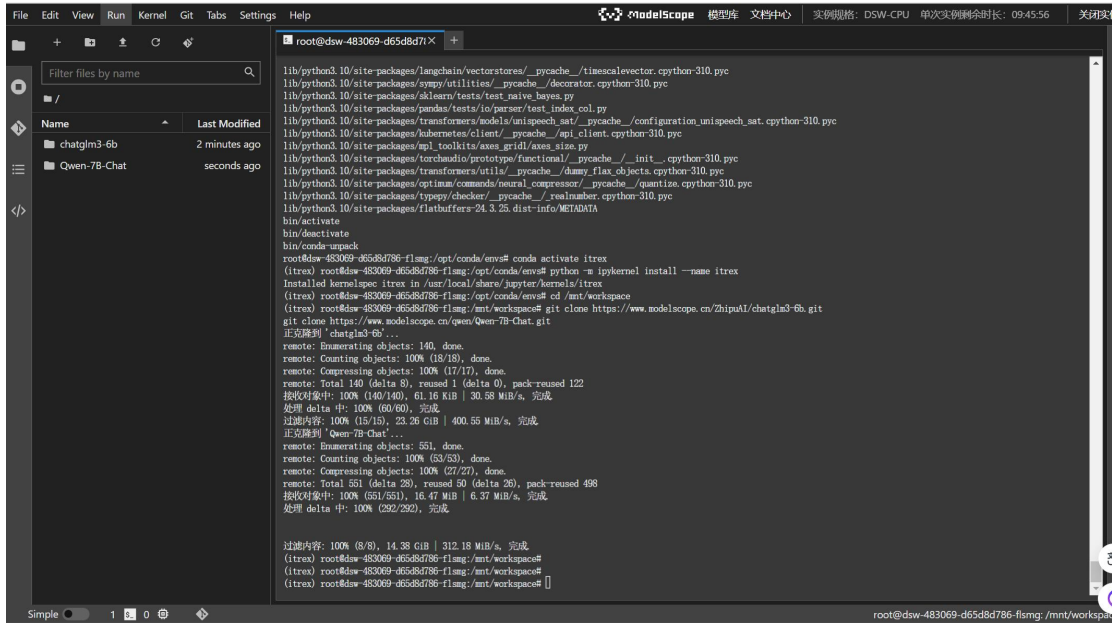
1.配置和部署(此部分对应作业要求 1：完成 git clone 相关 git 的截图或部署完成的相关截图)

注册完成后按照指导手册的要求打开 Notebook：



环境搭建的过程和完成配置成功：



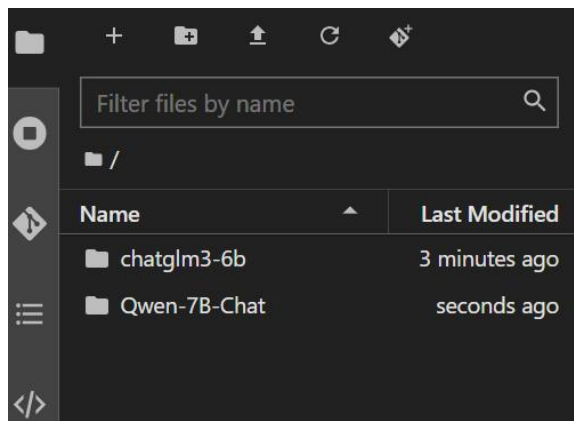


根据实验需要下载相对应的中文大模型至本地(完成 git clone 相关 git 的截图或部署完成的相关截图):

```
(itrex) root@dsw-483069-d65d8d786-flsmg:/mnt/workspace# git clone https://www.modelscope.cn/ZhipuAI/chatglm3-6b.git
git clone https://www.modelscope.cn/qwen/Qwen-7B-Chat.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 | 30.58 MiB/s, 完成
处理 delta 中: 100% (60/60), 完成
过滤内容: 100% (15/15), 23.26 GiB | 400.55 MiB/s, 完成
正克隆到 'Qwen-7B-Chat'...
remote: Enumerating objects: 551, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 551 (delta 28), reused 50 (delta 26), pack-reused 498
接收对象中: 100% (551/551), 16.47 MiB | 6.37 MiB/s, 完成
处理 delta 中: 100% (292/292), 完成

过滤内容: 100% (8/8), 14.38 GiB | 312.18 MiB/s, 完成
(itrex) root@dsw-483069-d65d8d786-flsmg:/mnt/workspace#
(itrex) root@dsw-483069-d65d8d786-flsmg:/mnt/workspace#
(itrex) root@dsw-483069-d65d8d786-flsmg:/mnt/workspace#
```

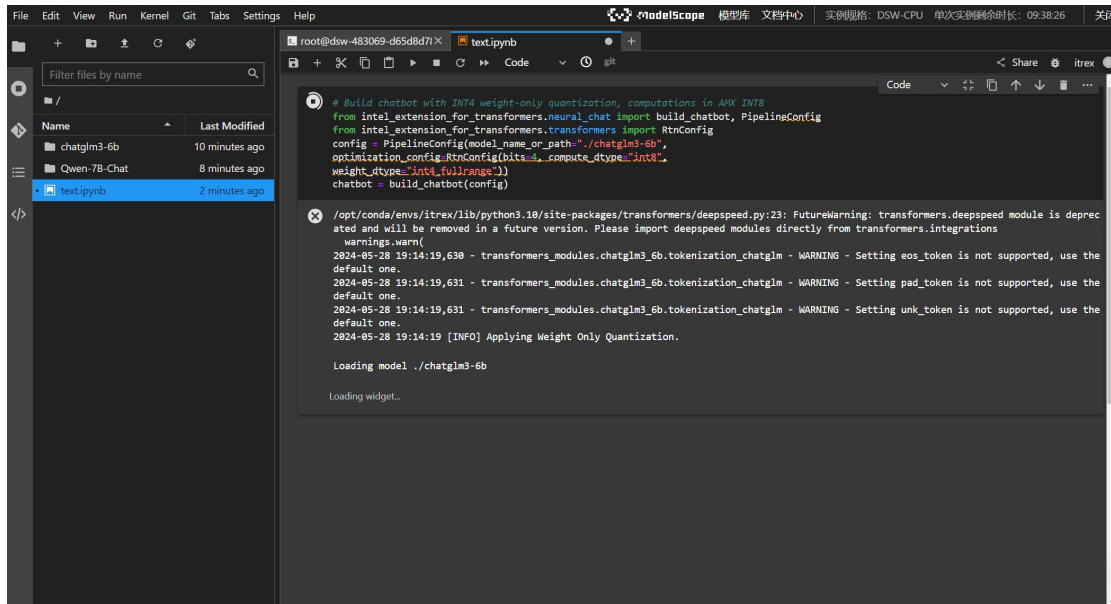
此时便可以在左侧目录下看到下载的两个大语言模型了。



2.对话与分析(此部分对应作业要求 2.3：问答测试结果的相关截图和语言模型之间的横向对比分析)

2.1.量化并构建 chatbot

使用智谱 ChatGLM3-6B 大语言模型：

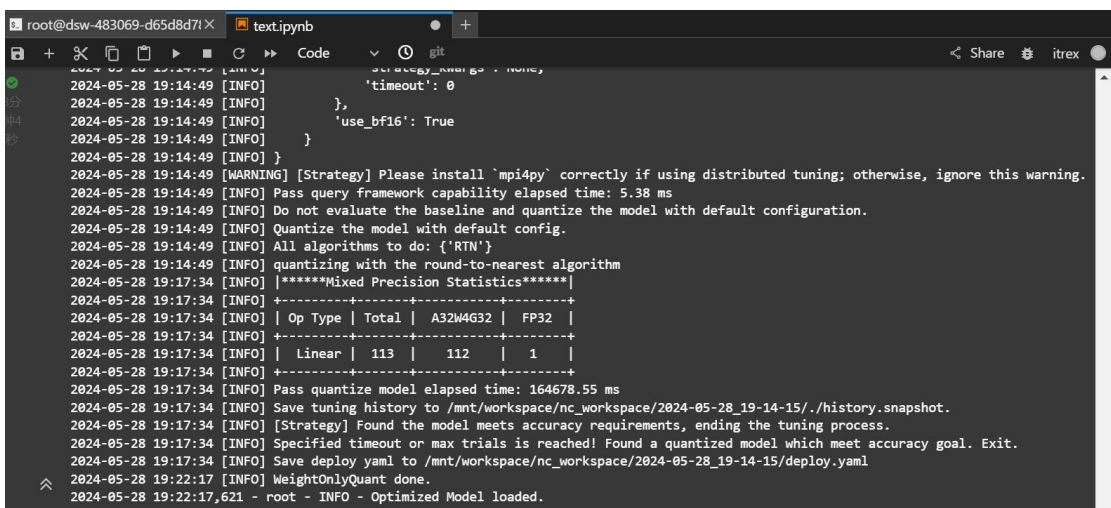


```
# Build chatbot with INT4 weight-only quantization, computations in AMX INT8
from intel_extension_for_transformers.neural_chat import build_chatbot, PipelineConfig
from intel_extension_for_transformers.transformers import RtnConfig
config = PipelineConfig(model_name_or_path="./chatglm3-6b",
                        optimization_config=RtnConfig(bits=4, compute_dtype="int8",
                                                    weight_dtype="int4_fullrange"))
chatbot = build_chatbot(config)

/opt/conda/envs/itrex/lib/python3.10/site-packages/transformers/deepspeed.py:23: FutureWarning: transformers.deepspeed module is deprecated and will be removed in a future version. Please import deepspeed modules directly from transformers.integrations
warnings.warn(
2024-05-28 19:14:19,630 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting eos_token is not supported, use the default one.
2024-05-28 19:14:19,631 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting pad_token is not supported, use the default one.
2024-05-28 19:14:19,631 - transformers_modules.chatglm3_6b.tokenization_chatglm - WARNING - Setting unk_token is not supported, use the default one.
2024-05-28 19:14:19 [INFO] Applying Weight Only Quantization.

Loading model ./chatglm3-6b

Loading widget...
```



```
2024-05-28 19:14:49 [INFO] strategy_kwargs: None,
2024-05-28 19:14:49 [INFO] 'timeout': 0
2024-05-28 19:14:49 [INFO] },
2024-05-28 19:14:49 [INFO] 'use_bf16': True
2024-05-28 19:14:49 [INFO] }
2024-05-28 19:14:49 [INFO] }
2024-05-28 19:14:49 [WARNING] [Strategy] Please install 'mpi4py' correctly if using distributed tuning; otherwise, ignore this warning.
2024-05-28 19:14:49 [INFO] Pass query framework capability elapsed time: 5.38 ms
2024-05-28 19:14:49 [INFO] Do not evaluate the baseline and quantize the model with default configuration.
2024-05-28 19:14:49 [INFO] Quantize the model with default config.
2024-05-28 19:14:49 [INFO] All algorithms to do: {'RTN'}
2024-05-28 19:14:49 [INFO] quantizing with the round-to-nearest algorithm
2024-05-28 19:17:34 [INFO] |*****Mixed Precision Statistics*****|
2024-05-28 19:17:34 [INFO] |-----+-----+-----+-----+|
2024-05-28 19:17:34 [INFO] | Op Type | Total | A32M4G32 | FP32 |
2024-05-28 19:17:34 [INFO] |-----+-----+-----+-----+|
2024-05-28 19:17:34 [INFO] | Linear | 113 | 112 | 1 |
2024-05-28 19:17:34 [INFO] |-----+-----+-----+-----+|
2024-05-28 19:17:34 [INFO] Pass quantize model elapsed time: 164678.55 ms
2024-05-28 19:17:34 [INFO] Save tuning history to /mnt/workspace/nc_workspace/2024-05-28_19-14-15/./history.snapshot.
2024-05-28 19:17:34 [INFO] [Strategy] Found the model meets accuracy requirements, ending the tuning process.
2024-05-28 19:17:34 [INFO] Specified timeout or max trials is reached! Found a quantized model which meet accuracy goal. Exit.
2024-05-28 19:17:34 [INFO] Save deploy yaml to /mnt/workspace/nc_workspace/2024-05-28_19-14-15/deploy.yaml
2024-05-28 19:22:17 [INFO] WeightOnlyQuant done.
2024-05-28 19:22:17,621 - root - INFO - Optimized Model loaded.
```

当我们想要换一个大语言模型时，将另一个大语言模型的路径复制换到相应的位置，再次量化并构建，下面就是换为通义千问 Qwen-7B-Chat 大语言模型的过程截图：

```
root@dsw-483069-d65d8d71X textipyb
# Build chatbot with INT4 weight-only quantization, computations in AMX INT8
from intel_extension_for_transformers.neural_chat import build_chatbot, PipelineConfig
from intel_extension_for_transformers.transformers import RtnConfig
config = PipelineConfig(model_name_or_path="./Qwen-7B-Chat",
                        optimization_config=RtnConfig(bits=4, compute_dtype="int8",
                                                    weight_dtype="int4_fullrange"))
chatbot = build_chatbot(config)

/opt/conda/envs/itrex/lib/python3.10/site-packages/transformers/deepspeed.py:23: FutureWarning: transformers.deepspeed module is deprecated and will be removed in a future version. Please import deepspeed modules directly from transformers.integrations
warnings.warn(

Loading model ./Qwen-7B-Chat

2024-05-28 19:33:30 [INFO] Applying Weight Only Quantization.

Loading widget...
```

```
ModelScope 模型库 文档中心 实例规格: DSW-CPU 单次实例剩余时长: 09:11:24
root@dsw-483069-d65d8d71X textipyb
2024-05-28 19:33:56 [INFO] }
2024-05-28 19:33:56 [WARNING] [Strategy] Please install 'mpi4py' correctly if using distributed tuning; otherwise, ignore this warning.
2024-05-28 19:33:56 [INFO] Pass query framework capability elapsed time: 3.96 ms
2024-05-28 19:33:56 [INFO] Do not evaluate the baseline and quantize the model with default configuration.
2024-05-28 19:33:56 [INFO] Quantize the model with default config.
2024-05-28 19:33:56 [INFO] All algorithms to do: {'RTN'}
2024-05-28 19:33:56 [INFO] quantizing with the round-to-nearest algorithm
2024-05-28 19:36:03 [INFO] |*****Mixed Precision Statistics*****|
2024-05-28 19:36:03 [INFO] +-----+-----+-----+-----+
2024-05-28 19:36:03 [INFO] | Op Type | Total | A32M4G32 | FP32 |
2024-05-28 19:36:03 [INFO] +-----+-----+-----+-----+
2024-05-28 19:36:03 [INFO] | Linear | 161 | 160 | 1 |
2024-05-28 19:36:03 [INFO] +-----+-----+-----+-----+
2024-05-28 19:36:03 [INFO] Pass quantize model elapsed time: 126462.33 ms
2024-05-28 19:36:03 [INFO] Save tuning history to /mnt/workspace/nc_workspace/2024-05-28_19-33-25/./history.snapshot.
2024-05-28 19:36:03 [INFO] [Strategy] Found the model meets accuracy requirements, ending the tuning process.
2024-05-28 19:36:03 [INFO] Specified timeout or max trials is reached! Found a quantized model which meet accuracy goal. Exit.
2024-05-28 19:36:03 [INFO] Save deploy yaml to /mnt/workspace/nc_workspace/2024-05-28_19-33-25/deploy.yaml
2024-05-28 19:40:53 [INFO] WeightOnlyQuant done.
2024-05-28 19:40:53,233 - root - INFO - Optimized Model loaded.
```

2.2.对话测试和横向分析比较

问题一.请说出以下两句话区别在哪里?1.冬天：能穿多少穿多少；2.夏天：能穿多少穿多少；

通义千问 Qwen-7B-Chat 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少")
print(response)

/opt/conda/envs/itrex/lib/python3.10/site-packages/torch/amp/autocast_mode.py:267: UserWarning: In CPU autocast, but the target dtype is not supported. Disabling autocast.
CPU Autocast only supports dtype of torch.bfloat16, torch.float16 currently.
warnings.warn(error_message)

这两句话的区别在于它们描述的季节不同。第一句话“冬天：能穿多少穿多少”暗示着在寒冷的冬季，人们需要尽可能地穿上衣服来保暖；而第二句话“夏天：能穿多少穿多少”则表示在炎热的夏季，人们也需要尽可能地减少衣物以保持凉爽。因此，这两句话所表达的意思是相反的。
```

智谱 ChatGLM3-6B 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="请说出以下两句话区别在哪里？ 1、冬天：能穿多少穿多少 2、夏天：能穿多少穿多少")
print(response)

/opt/conda/envs/itrex/lib/python3.10/site-packages/torch/amp/autocast_mode.py:267: UserWarning: In CPU autocast, but the target dtype is not supported. Disabling autocast.
CPU Autocast only supports dtype of torch.bfloat16, torch.float16 currently.
warnings.warn(error_message)

这两句话的意思是相同的，都是在询问在寒冷的冬天和炎热的夏天，人们应该穿多少衣服。但是，冬天和夏天的气候条件不同，所以人们在选择穿什么衣服上有不同的考虑。冬天通常很冷，所以需要穿厚实的外套、保暖内衣和舒适的鞋子等；而夏天则很热，人们需要穿轻便的衣服、短裤和凉鞋等。因此，虽然这两句话的意思相同，但它们所反映的气候条件和人们对穿着的需求不同。
```


对于第一个问题，我们人类的正常理解是前半句的意思是冬天要穿多，夏天要穿少，两句的意思相反；通义千问 Qwen-7B-Chat 的回答和人类的正常思维相符；智谱 ChatGLM3-6B 的回答虽然说两句话意思相同，但是后面的解释也与人类思维大差不差。再考虑两者回答所需时间，都是一分零 2 秒。所以综上所述，对于第一个问题，两种大语言模型都很好地做出了正确的回答。

问题二.请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上；

通义千问 Qwen-7B-Chat 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上")
print(response)
```

这两句话的区别在于它们的语境和含义不同。第一句话是在描述一种现象，即有些人因为找不到合适的伴侣而被称为“单身狗”，并解释了产生这种现象的原因是“谁都看不上”。第二句话则是在表达一个人的观点或态度，即认为自己之所以成为“单身狗”是因为“谁都看不上”，这是一种主观的看法。因此，虽然这两句话都提到了“单身狗”和“谁都看不上”的概念，但它们的语境和含义有所不同。

智谱 ChatGLM3-6B 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="请说出以下两句话区别在哪里？单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上")
print(response)
```

这两句话的语义重复。第一句“单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上”表达的意思是单身的原因有两个，一个是看不上别人，另一个是别人看不上自己。第二句“单身狗产生的原因有两个，一是谁都看不上，二是谁都看不上”与第一句意思相同，只是表述方式略有不同。因此，这两句话在语义上存在重复。

对于第二个问题，我们人类的正常理解是人单身的原因，可能是自己看不上他人，也有可能是他人看不上自己。通义千问 Qwen-7B-Chat 的回答与智谱 ChatGLM3-6B 的回答在解释上都和人类的正常思维相符，但是 ChatGLM3-6B 回答中的两句话语意重复我有点没看明白，可能是暂时性出错吧。再考虑两者回答所需时间，通义千问 Qwen-7B-Chat 是一分零二秒，ChatGLM3-6B 是 52 秒。综上所述，通义千问 Qwen-7B-Chat 回答的更加容易理解、准确，ChatGLM3-6B 虽然回答快些，但是出了点不影响整体意思的小问题。

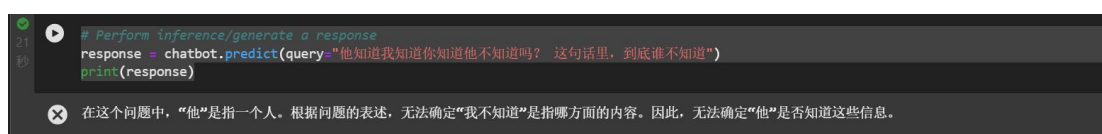
问题三.他知道我知道你知道他不知道吗？这句话里，到底谁不知道；

通义千问 Qwen-7B-Chat 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="他知道我知道你知道他不知道吗？这句话里，到底谁不知道")
print(response)
```

这句话的含义是：这个人知道我知道你知道他不知道。因此，这个人的状态是知道我们知道他知道他不知道。

智谱 ChatGLM3-6B 回答:



```
# Perform inference/generate a response
response = chatbot.predict(query="他知道我知道他知道他不知道吗？这句话里，到底谁不知道")
print(response)
```

在这个问题中，“他”是指一个人。根据问题的表述，无法确定“我不知道”是指哪方面的内容。因此，无法确定“他”是否知道这些信息。

对于第三个问题，我们人类的正常理解应该是“他”不知道，这句话大致意思是，“你”知道“他不知道”这一事实，我又知道“你知道他不知道”这一事实，“我”想问“他”到底知不知道“我知道你知道他不知道”这一事实。通义千问 Qwen-7B-Chat 的回答没有给出很好的解释，而且给出的答案是错的，因为他并不知道；智谱 ChatGLM3-6B 也没回答上来，而且感觉答非所问。再考虑两者回答所需时间，两者都是 21 秒。综上所述，两者对这道题的回答均不理想，但是我决定通义千问 Qwen-7B-Chat 回答地更深入些，因为它回答了“他”到底知不知道这一问题，虽然最后回答错了；智谱 ChatGLM3-6B 回答的更不着边际，就是它肯还没懂这句话是在问它什么。

问题四.明明明明明白白白喜欢他，可她就是不说话。这句话里，明明和白白谁喜欢谁？

通义千问 Qwen-7B-Chat 回答:



```
# Perform inference/generate a response
response = chatbot.predict(query="明明明明明白白白喜欢他，可她就是不说话。这句话里，明明和白白谁喜欢谁？")
print(response)
```

根据句子的描述，“明明”喜欢“他”，而“白白”没有明确表示自己喜欢谁。因此，可以得出结论：明明喜欢他。

智谱 ChatGLM3-6B 回答:



```
# Perform inference/generate a response
response = chatbot.predict(query="明明明明明白白白喜欢他，可她就是不说话。这句话里，明明和白白谁喜欢谁？")
print(response)
```

根据句子中的信息，无法确定明明和白白谁喜欢谁。因为句子中并没有明确说明他们的感情状态。

对于第四个问题，我们人类的正常理解应该是白白喜欢明明。我们从句子中可以知道明明是男生，白白是女生，白白喜欢明明但是白白没有说出口。通义千问 Qwen-7B-Chat 的回答是错的；智谱 ChatGLM3-6B 的回答感觉又犯了前面一个问题的错误，就是没有读懂我在问什么，它连谁喜欢谁都没有做出判断。再考虑两者回答所需时间，通义千问 Qwen-7B-Chat 是 27 秒，智谱 ChatGLM3-6B 是 15 秒。综上所述，我觉得依旧是通义千问 Qwen-7B-Chat 回答更好，虽然他回答

错误，但是它至少是在回答这个问题的具体内容；但是智谱 ChatGLM3-6B 直接连判断都做不出来了。

问题五.领导：你这是什么意思?小明：没什么意思。意思意思。领导：你这就不够意思了。小明：小意思，小意思。领导：你这人真有意思。小明：其实也没有别的意思。领导：那我就不好意思了。小明：是我不好意思。请问：以上“意思”分别是什么意思。

通义千问 Qwen-7B-Chat 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="领导：你这是什么意思？ 小明：没什么意思。意思意思。 领导：你这就不够意思了。 小明：小意思，小意思。领导：你这人真有意思。小明：其实也没有别的意思。领导：那我就不好意思了。小明：是我不好意思。请问：以上“意思”分别是什么意思。")
print(response)
```

根据上下文，我们可以推断出以下含义：

1. “没什么意思”：表示没有特别的意思或想法。
2. “意思意思”：表示有些意思或想法，但不是很明确或详细。
3. “不够意思”：表示没有达到预期的标准或要求。
4. “小意思”：表示一些微不足道的事情或贡献。
5. “有意思”：表示有一些有趣、引人入胜的事情或情况。
6. “其实也没有别的意思”：表示实际上并没有其他的意思或意图。
7. “那我就不好意思了”：表示因为某些原因感到尴尬或难堪。
8. “是我不好意思”：表示因为自己的行为或言语而感到抱歉或羞愧。

因此，以上对话中的“意思”可以理解为不同的含义和情绪状态。

智谱 ChatGLM3-6B 回答：

```
# Perform inference/generate a response
response = chatbot.predict(query="领导：你这是什么意思？ 小明：没什么意思。意思意思。 领导：你这就不够意思了。 小明：小意思，小意思。领导：你这人真有意思。小明：其实也没有别的意思。领导：那我就不好意思了。小明：是我不好意思。请问：以上“意思”分别是什么意思。")
print(response)
```

1. “意思”第一层含义是指对某种事物的理解或解释。
2. “意思”第二层含义是指对某种行为或语言的隐喻或象征性的解释。
3. “意思”第三层含义是指对某种事物或行为的实际意义或作用的理解。

对于第五个问题，确实有点绕，以下是我作为人类的理解：领导一开始问小明行为的意图(第一个意思)，小明回答没啥重要的(第二个意思)，只是想表达一下对领导的感激(第三个意思意思)，领导回答小明这行为可能不太好(第四个意思)，小明回答这只是小问题(比如送小礼物啥的，第五个意思)，领导回复小明这人不错，懂得找领导套近乎关系(第六个意思)，小明回答其实也只是想感谢一下领导，没有想叫领导帮忙照顾啥的(第七个意思)，领导回复那他这样白白收礼但不帮助小明他不太好意思(第八个意思)，小明谦称说领导不要不好意思是我自己送礼不好意思。通义千问 Qwen-7B-Chat 的回答是挺不错的，每一个“意思”的意思都解释得较为清楚而且与人类思维大致相符；智谱 ChatGLM3-6B 的回答就差点意思，只是笼统地解释“意思”这个词在众多语境中的意思，没有代入具体的分析，好像它没看懂问题的文本一样。再考虑两者回答所需时间，通义千问

Qwen-7B-Chat 是两分十五秒，智谱 ChatGLM3-6B 是 40 秒。综上所述，我认为通义千问 Qwen-7B-Chat 回答的更好，虽然它花点时间更多，但是回答质量更高。

3.总结分析(此部分对应作业要求 3：语言模型之间的横向对比分析)

通义千问-7B(Qwen-7B)是阿里云研发的通义千问大模型系列的 70 亿参数规模的模型。它是基于 Transformer 的大语言模型，在超大规模的预训练数据上进行训练得到。ChatGLM3 是智谱 AI 和清华大学 KEG 实验室联合发布的对话预训练模型。两者都是当代优秀的大语言模型。

在本次项目的五个问题中，两种大语言模型都各有所长。对于比较简单的问题(比如问题一和问题二)来说，两者都做出了较好的回答，但是智谱 ChatGLM3-6B 可能在一些小细节的表述上不太到位，但是整体的意思是正确的；对于中等难度的题目(比如问题三和问题四)来说，通义千问 Qwen-7B-Chat 虽然知道你在问什么，但是给出了错误的结果，ChatGLM3-6B 好像看不懂问题所在，没有给出确切的答案，而是给出“无法做出判断”类似这样的不确定性语句；对于难度较高的题目(比如问题五)，通义千问 Qwen-7B-Chat 可以给出较为正确的答案，但是回答的速度并不快，ChatGLM3-6B 的回答就比较没有贴合实际的语境，给出的回答太过于笼统泛化，可能是它没有充分地利用上下文语境，没有很好地理解所问问题大意的关键所在。

两者都是优秀的大语言模型，经过这次项目实验，我给出基于个人开发的意见：通义千问 Qwen-7B-Chat 可能需要在问题回复正确性上多加训练改进；ChatGLM3-6B 可能需要在问题的深刻理解上多加训练改进。