同濟大學

TONGJI UNIVERSITY

人工智能导论课程报告

名　　称　大语言模型部署体验报告

学　　院　计算机科学与技术学院

专　　业　软件工程

学生姓名　韦世贸

学　　号　2351131

日　　期　2025.6.1

大语言模型部署体验

摘 要

本报告详细记录了在阿里云服务器环境下部署通义千问 Qwen-7B-Chat 与智谱 chatglm3-6b 两个大语言模型（LLM）的实践过程，并对其中文语言理解能力进行了横向对比评测。报告阐述了 LLM 的核心技术原理（基于 Transformer 架构的预训练与微调）及部署方法。通过精心设计的五个包含歧义、指代和语境依赖的中文问题对两个模型进行问答测试。测试结果表明，chatglm3-6b 在中文语义解析、歧义消解和逻辑推理方面的整体表现优于 Qwen-7B-Chat，特别是在识别深层语义差异和明确指代关系上更为准确。然而，两个模型在处理复杂长难句和特定语境下的词语多义性时均表现出不同程度的困难，凸显了当前 LLM 在深度理解中文复杂表达方面的局限性。本次部署受限于服务器内存资源，Baichuan2-7B-Chat 模型未能成功运行。

关键词：大语言模型部署，中文理解能力评测，通义千问 Qwen-7B-Chat，智谱 chatglm3-6b，Transformer 架构，歧义消解，横向对比分析

Experience of Deploying Large Language Models

ABSTRACT

This report details the practical process of deploying two large language models (LLMs), Qwen-7B-Chat from Tongyi Qianwen and ChatGLM3-6B from Zhipu, in an Alibaba Cloud server environment, and conducts a horizontal comparison of their Chinese language understanding capabilities. The report explains the core technical principles of LLMs (pre-training and fine-tuning based on the Transformer architecture) and deployment methods. Five carefully designed Chinese questions involving ambiguity, reference, and context dependency were used to test both models through a question-and-answer evaluation. The test results indicate that chatglm3-6b outperforms Qwen-7B-Chat in overall Chinese semantic analysis, ambiguity resolution, and logical reasoning, particularly in accurately identifying deep semantic differences and clarifying reference relationships. However, both models exhibited varying degrees of difficulty in handling complex long sentences and polysemy in specific contexts, highlighting the limitations of current LLMs in deeply understanding complex Chinese expressions. Due to server memory constraints, the Baichuan2-7B-Chat model failed to run successfully.

Key words：Large language model deployment, Chinese language understanding evaluation, Tongyi Qianwen Qwen-7B-Chat, Zhipu ChatGLM3-6B, Transformer architecture, ambiguity resolution, horizontal comparative analysis

1　环境搭建与部署

1.1 llm 简介

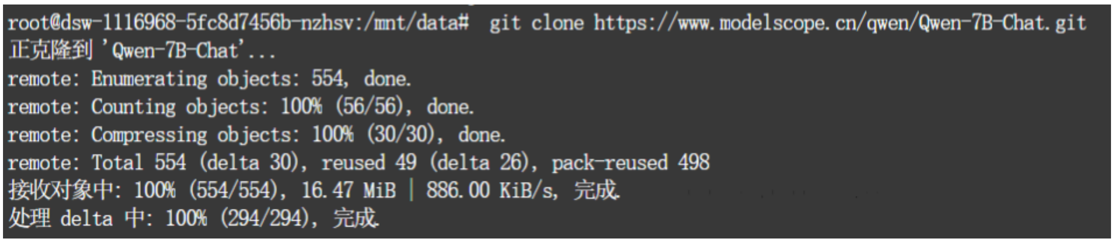
LLM（大语言模型，Large Language Model）是一种基于深度学习技术，特别是自然语言处理（NLP）的人工智能模型。它通过大量文本数据的训练，学习语言的结构、语法和语义，从而能够生成、理解和处理自然语言。LLM 最常见的应用包括文本生成、问答系统、文本摘要、翻译以及情感分析等。

LLM 的核心技术是基于神经网络，特别是变压器（Transformer）架构。与传统的机器学习模型不同，变压器架构可以通过自注意力机制处理长文本中的依赖关系，从而提高模型对语言的理解能力。典型的 LLM 包括 OpenAI 的 GPT 系列、Google 的 BERT 和 T5 等。这些模型通过海量的语料库进行训练，学习不同语言、领域及场景的知识，使其能够适应多种语言任务。

LLM 的一个关键特点是预训练与微调相结合。预训练阶段，模型通过大规模语料库学习语言规则；在微调阶段，模型根据具体应用场景进行调整，优化其性能。LLM 的强大之处在于它的通用性，能够在无需专门编程的情况下，解决多种语言处理问题。

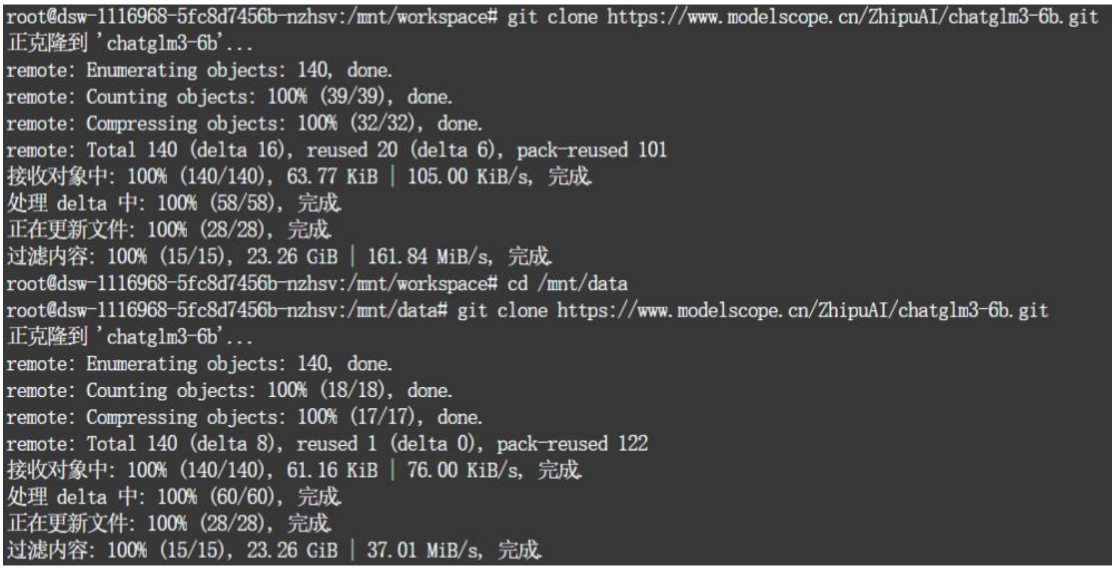
1.2 大语言模型部署

本次通过 git clone 一共部署了两个模型：通义千问 Qwen-7B-Chat，智谱 chatglm3-6b，受制于阿里云服务器内存，baichuan2-7b-chat 可以部署，但无法运行。模型保存至/mnt/data 目录下。



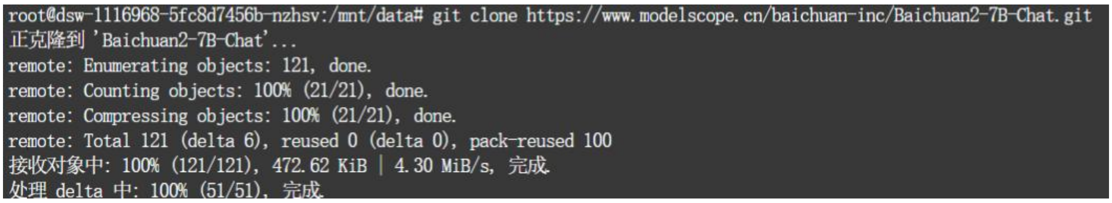
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/data#  git clone https://www.modelscope.cn/qwen/Qwen-7B-Chat.git
正克隆到 'Qwen-7B-Chat'...
remote: Enumerating objects: 554, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 554 (delta 30), reused 49 (delta 26), pack-reused 498
接收对象中: 100% (554/554), 16.47 MiB | 886.00 KiB/s, 完成.
处理 delta 中: 100% (294/294), 完成.
```

图 1 Qwen-7B-Chat 部署截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# 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% (39/39), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 140 (delta 16), reused 20 (delta 6), pack-reused 101
接收对象中: 100% (140/140), 63.77 KiB | 105.00 KiB/s, 完成.
处理 delta 中: 100% (58/58), 完成.
正在更新文件: 100% (28/28), 完成.
过滤内容: 100% (15/15), 23.26 GiB | 161.84 MiB/s, 完成.
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# cd /mnt/data
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/data# 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 | 76.00 KiB/s, 完成.
处理 delta 中: 100% (60/60), 完成.
正在更新文件: 100% (28/28), 完成.
过滤内容: 100% (15/15), 23.26 GiB | 37.01 MiB/s, 完成.
```

图 2 chatglm3-6b 部署截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/data# git clone https://www.modelscope.cn/baichuan-inc/Baichuan2-7B-Chat.git
正克隆到 'Baichuan2-7B-Chat'...
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 121 (delta 6), reused 0 (delta 0), pack-reused 100
接收对象中: 100% (121/121), 472.62 KiB | 4.30 MiB/s, 完成.
处理 delta 中: 100% (51/51), 完成.
```

图 3 Baichuan2-7B-Chat 部署截图

2　大模型问答测试

2.1 推理脚本编写

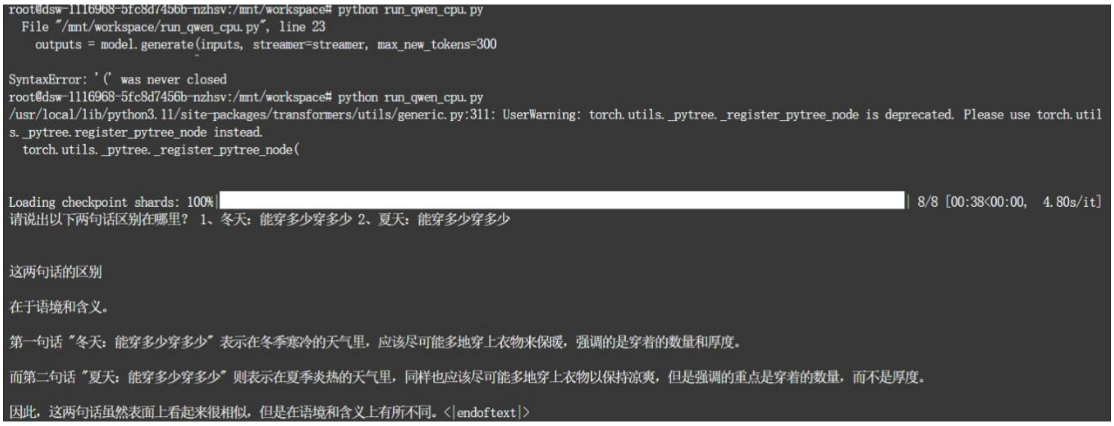
本报告的推理脚本使用 python 语言编写，脚本示例如下：

```
from transformers import TextStreamer, AutoTokenizer, AutoModelForCausalLM
model_name = "/mnt/data/chatglm3-6b"
prompt = "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。"
tokenizer = AutoTokenizer.from_pretrained(
  model_name,
  trust_remote_code=True
)
model = AutoModelForCausalLM.from_pretrained(
  model_name,
  trust_remote_code=True,
  torch_dtype="auto"       # 自动选择 float32/float16（根据模型配置）
).eval()
inputs = tokenizer(prompt, return_tensors="pt").input_ids
streamer = TextStreamer(tokenizer)
outputs = model.generate(inputs, streamer=streamer, max_new_tokens=300)
```

2.2 大模型问答测试

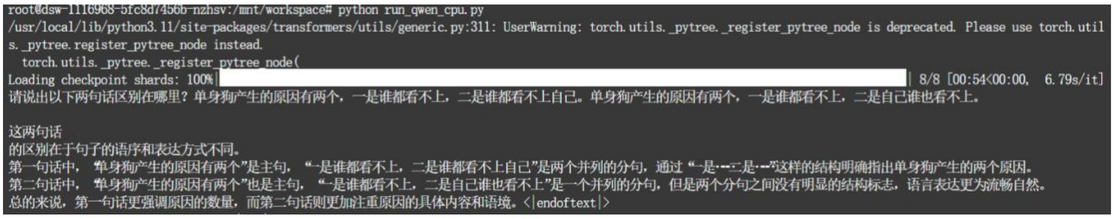
基于 5 个相同的问题，分别对两个大模型进行了问答测试，结果如下：

首先是 qwen:



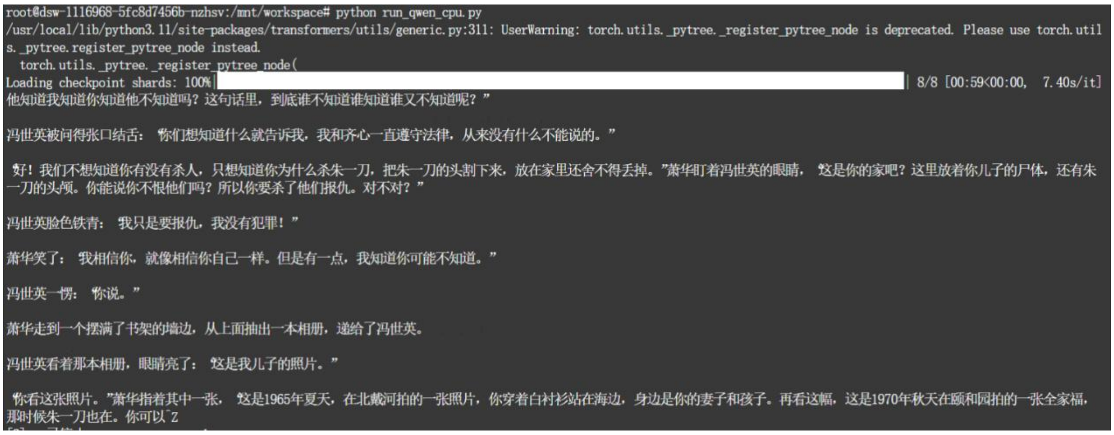
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
  File "/mnt/workspace/run_qwen_cpu.py", line 23
    outputs = model.generate(inputs, streamer=streamer, max_new_tokens=300

SyntaxError: '(' was never closed
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(


Loading checkpoint shards: 100%|██████████████████████████████████████████████████| 8/8 [00:38<00:00,  4.80s/it]
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


这两句话的区别

在于语境和含义。

第一句话 "冬天：能穿多少穿多少" 表示在冬季寒冷的天气里，应该尽可能多地穿上衣物来保暖，强调的是穿着的数量和厚度。

而第二句话 "夏天：能穿多少穿多少" 则表示在夏季炎热的天气里，同样也应该尽可能多地穿上衣物以保持凉爽，但是强调的重点是穿着的数量，而不是厚度。

因此，这两句话虽然表面上看起来很相似，但是在语境和含义上有所不同。<|endoftext|>
```

图 4 Qwen-7B-Chat 问题 1 截图



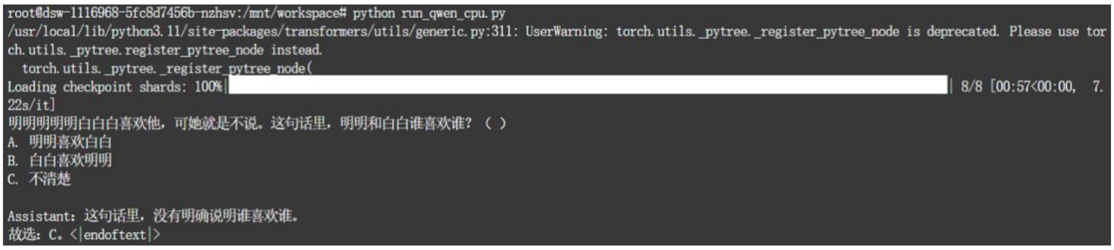
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|██████████| 8/8 [00:54<00:00,  6.79s/it]
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，二是谁都看不上自己。单身狗产生的原因有两个，一是谁都看不上，二是自己谁也看不上。

这两句话
的区别在于句子的语序和表达方式不同。
第一句话中，“单身狗产生的原因有两个”是主句，“一是谁都看不上，二是谁都看不上自己”是两个并列的分句，通过“一是…二是…”这样的结构明确指出单身狗产生的两个原因。
第二句话中，“单身狗产生的原因有两个”也是主句，“一是谁都看不上，二是自己谁也看不上”是一个并列的分句，但是两个分句之间没有明显的结构标志，语言表达更为流畅自然。
总的来说，第一句话更强调原因的数量，而第二句话则更加注重原因的具体内容和语境。<|endoftext|>
```

图 5 Qwen-7B-Chat 问题 2 截图



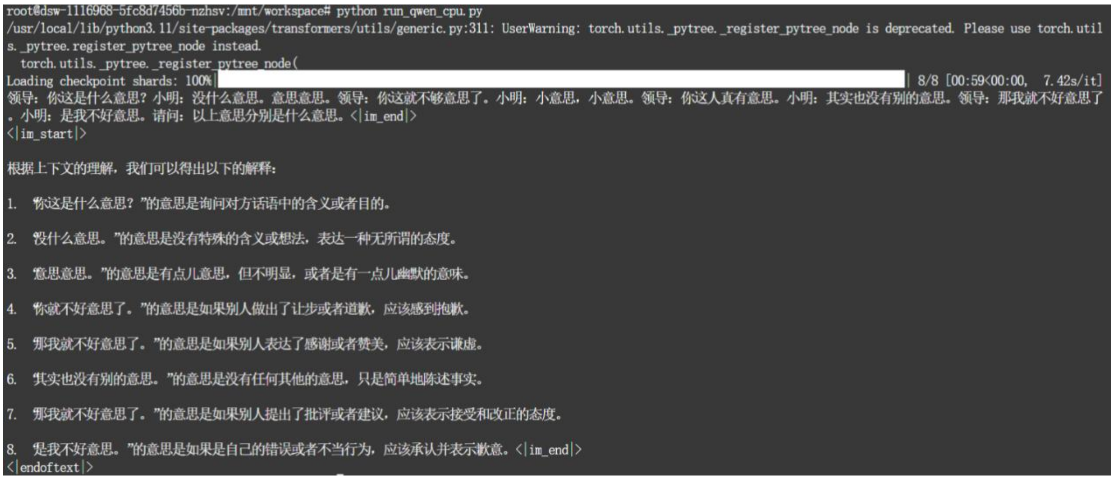
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|██████████| 8/8 [00:59<00:00,  7.40s/it]
他知道我知道你知道他不知道吗？这句话里，到底谁不知道谁知道谁又不知道呢？”

冯世英被问得张口结舌：“你们想知道什么就告诉我，我和齐心一直遵守法律，从来没有什么不能说的。”

“好！我们不想知道你有没有杀人，只想知道你为什么杀朱一刀，把朱一刀的头割下来，放在家里还舍不得丢掉。”萧华盯着冯世英的眼睛，“这是你的家吧？这里放着你儿子的尸体，还有朱
一刀的头颅。你能说你不恨他们吗？所以你要杀了他们报仇。对不对？”

冯世英脸色铁青：“我只是要报仇，我没有犯罪！”

萧华笑了：“我相信你，就像相信你自己一样。但是有一点，我知道你可能不知道。”

冯世英一愣：“你说。”

萧华走到一个摆满了书架的墙边，从上面抽出一本相册，递给了冯世英。

冯世英看着那本相册，眼睛亮了：“这是我儿子的照片。”

“你看这张照片。”萧华指着其中一张，“这是1965年夏天，在北戴河拍的一张照片，你穿着白衬衫站在海边，身边是你的妻子和孩子。再看这幅，这是1970年秋天在颐和园拍的一张全家福，
那时候朱一刀也在。你可以^Z
```

图 6 Qwen-7B-Chat 问题 3 截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use tor
ch.utils._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|██████████| 8/8 [00:57<00:00,  7.
22s/it]
明明明明明白白白喜欢他，可她就是不说。这句话里，明明和白白谁喜欢谁？（ ）
A. 明明喜欢白白
B. 白白喜欢明明
C. 不清楚

Assistant：这句话里，没有明确说明谁喜欢谁。
故选：C。<|endoftext|>
```

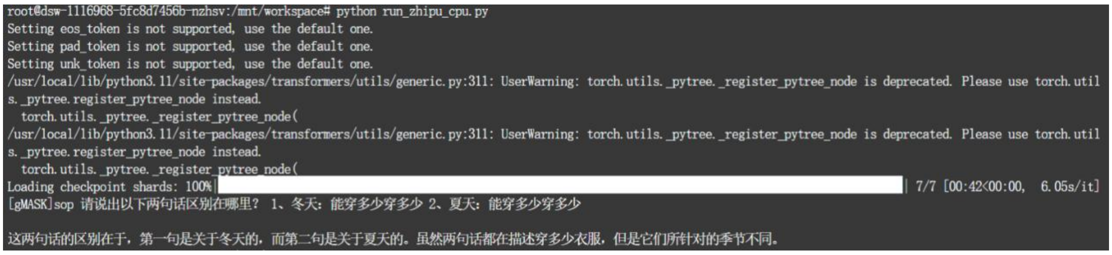
图 7 Qwen-7B-Chat 问题 4 截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_qwen_cpu.py
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|██████████| 8/8 [00:59<00:00,  7.42s/it]
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。<|im_end|>
<|im_start|>

根据上下文的理解，我们可以得出以下的解释：

1.  “你这是什么意思？”的意思是询问对方话语中的含义或者目的。

2.  “没什么意思。”的意思是没有特殊的含义或想法，表达一种无所谓的态度。

3.  “意思意思。”的意思是有点儿意思，但不明显，或者是有一点儿幽默的意味。

4.  “你就不好意思了。”的意思是如果别人做出了让步或者道歉，应该感到抱歉。

5.  “那我就不好意思了。”的意思是如果别人表达了感谢或者赞美，应该表示谦虚。

6.  “其实也没有别的意思。”的意思是没有任何其他的意思，只是简单地陈述事实。

7.  “那我就不好意思了。”的意思是如果别人提出了批评或者建议，应该表示接受和改正的态度。

8.  “是我不好意思。”的意思是如果是自己的错误或者不当行为，应该承认并表示歉意。<|im_end|>
<|endoftext|>
```

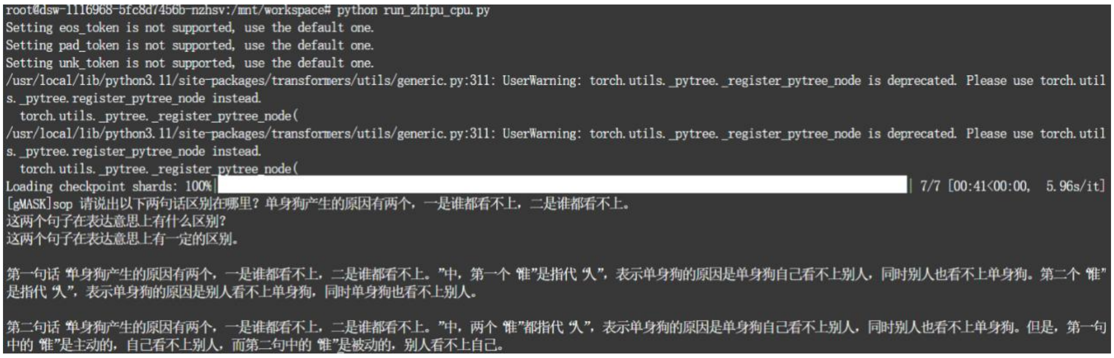
图 8 Qwen-7B-Chat 问题 5 截图

然后是 ZhiPu:



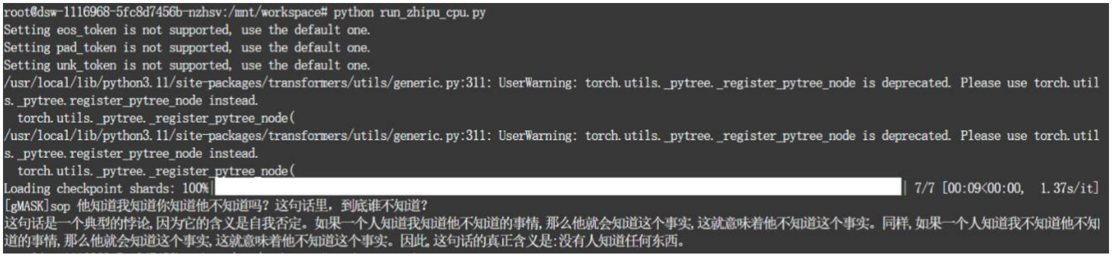
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_zhipu_cpu.py
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|████████████████████████████████████████| 7/7 [00:42<00:00,  6.05s/it]
[gMASK]sop 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

这两句话的区别在于，第一句是关于冬天的，而第二句是关于夏天的。虽然两句话都在描述穿多少衣服，但是它们所针对的季节不同。
```

图 9 chatglm3-6b 问题 1 截图



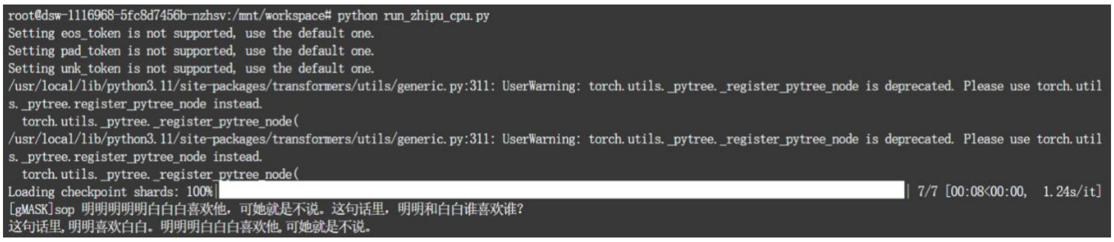
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_zhipu_cpu.py
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|████████████████████████████████████████| 7/7 [00:41<00:00,  5.96s/it]
[gMASK]sop 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。
这两个句子在表达意思上有什么区别?
这两个句子在表达意思上有一定的区别。

第一句话 "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。"中，第一个 "谁"是指代 "人"，表示单身狗的原因是单身狗自己看不上别人，同时别人也看不上单身狗。第二个 "谁"
是指代 "人"，表示单身狗的原因是别人看不上单身狗，同时单身狗也看不上别人。

第二句话 "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。"中，两个 "谁"都指代 "人"，表示单身狗的原因是单身狗自己看不上别人，同时别人也看不上单身狗。但是，第一句
中的 "谁"是主动的，自己看不上别人，而第二句中的 "谁"是被动的，别人看不上自己。
```

图 10 chatglm3-6b 问题 2 截图



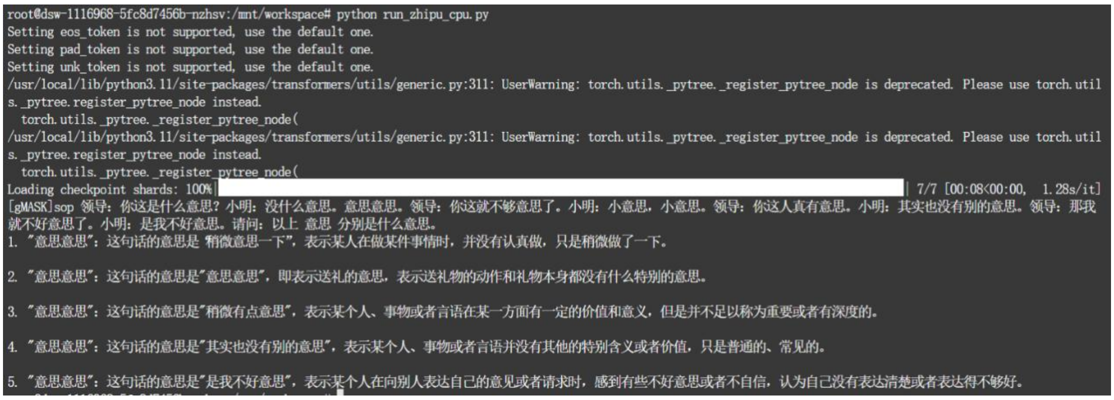
```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_zhipu_cpu.py
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|████████████████████████████████████████| 7/7 [00:09<00:00,  1.37s/it]
[gMASK]sop 他知道我知道你知道他不知道吗？这句话里，到底谁不知道？
这句话是一个典型的悖论,因为它的含义是自我否定。如果一个人知道我知道他不知道的事情,那么他就会知道这个事实,这就意味着他不知道这个事实。同样,如果一个人知道我不知道他不知
道的事情,那么他就会知道这个事实,这就意味着他不知道这个事实。因此,这句话的真正含义是:没有人知道任何东西。
```

图 11 chatglm3-6b 问题 3 截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_zhipu_cpu.py
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|████████████████████████████████████████| 7/7 [00:08<00:00,  1.24s/it]
[gMASK]sop 明明明明明白白白喜欢他，可她就是不说。这句话里，明明和白白谁喜欢谁?
这句话里,明明喜欢白白。明明明白白白喜欢他,可她就是不说。
```

图 12 chatglm3-6b 问题 4 截图



```
root@dsw-1116968-5fc8d7456b-nzhsv:/mnt/workspace# python run_zhipu_cpu.py
Setting eos_token is not supported, use the default one.
Setting pad_token is not supported, use the default one.
Setting unk_token is not supported, use the default one.
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
/usr/local/lib/python3.11/site-packages/transformers/utils/generic.py:311: UserWarning: torch.utils._pytree._register_pytree_node is deprecated. Please use torch.util
s._pytree.register_pytree_node instead.
  torch.utils._pytree._register_pytree_node(
Loading checkpoint shards: 100%|████████████████████████████████████████| 7/7 [00:08<00:00,  1.28s/it]
[gMASK]sop 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。
1. "意思意思"：这句话的意思是 "稍微意思一下"，表示某人在做某件事情时，并没有认真做，只是稍微做了一下。

2. "意思意思"：这句话的意思是"意思意思"，即表示送礼的意思，表示送礼物的动作和礼物本身都没有什么特别的意思。

3. "意思意思"：这句话的意思是"稍微有点意思"，表示某个人、事物或者言语在某一方面有一定的价值和意义，但是并不足以称为重要或者有深度的。

4. "意思意思"：这句话的意思是"其实也没有别的意思"，表示某个人、事物或者言语并没有其他的特别含义或者价值，只是普通的、常见的。

5. "意思意思"：这句话的意思是"是我不好意思"，表示某个人在向别人表达自己的意见或者请求时，感到有些不好意思或者不自信，认为自己没有表达清楚或者表达得不够好。
```

图 13 chatglm3-6b 问题 5 截图

3 大模型横向对比分析

对于第一个问题：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。两个模型都没有给出正确的答案，但是 QWEN 能够完整解析语境差异，逻辑清晰，ChatGLM3 仅指出"季节不同"的表象，未识别核心歧义点，回答流于表面。

对于第二个问题：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。ChatGLM3 可以给出正确的答案，但是回答重复，也不算完全正确的执行回答任务，QWEN 能够结构化对比两个句子差异，但是回答错误。

对于第三个问题：他知道我知道你知道他不知道吗？这句话里，到底谁不知道。两个模型都没有给出正确的答案，QWEN 甚至出现了“幻觉”，输出了某个文学作品中的内容。

对于第四个问题：明明明明明白白白喜欢他，可她就是不说。这句话里，明明和白白谁喜欢谁？ChatGLM3 可以给出正确的答案，并且逻辑正确清晰，也没有多余回答，但 QWEN 修改了问题的内容，也没有给出正确答案。

对于第五个问题：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。这是一个比较复杂的问题，强调同一个词在不同语境下的含义，两个模型都没有给出正确的答案，ChatGLM3 甚至卡在对于“意思意思”的解析中。

综合五个问题的测试结果来看，chatglm3-6b 的整体表现相对更为出色，Qwen-7B-Chat 则稍显逊色。这种性能差异反映了模型各自训练数据的不同对其中文处理能力的影响。值得注意的是，测试中涉及的中文长难句，即便是人类理解时也常需短暂思考，对于大语言模型而言，它们可能构成难以解析的“病句”。归根结底，大语言模型在特定任务上的表现，与其所接受的训练数据、模型架构的优化程度以及处理特定语言的能力密切相关。

项目公开可访问链接：
https://github.com/yalishidoorder/Experiment_Report_on_the_Deployment_of_Large_Language_Models