实验6： 预训练模型使用

**1. 实验目的**

掌握主流预训练模型的特点、调用及微调方法，了解主流的大模型，理解使用预训练模型实现NLP任务的基本流程。

**2. 实验平台**

操作系统：Windows 2000/ XP/7/8/10/11 或者 Linux

深度学习框架：pytorch

**3. 实验内容**

**任务：趣味科学问答**

**细节：查阅网上资源（学术论文、浏览器、大模型等）**

* 使用GPT-4、DeepSeek-v3或其他大规模的大模型，生成350条趣味科学问答数据（为保证效果需要的数据越多越好，如至少1000条，此处350仅为本次实验要求，并非保证本任务效果的最低要求）。随机划分300训练，50测试。问答数据示例：

Q：为什么雨后森林里会有蘑菇？

A：森林里的小蘑菇其实是一群躲雨的小精灵变出来的魔法伞！当下完雨后：

1. 泥土爷爷喝饱了水，变得软乎乎的，就像一块巨大的海绵蛋糕！
2. 蘑菇宝宝们早就躲在土里睡觉觉，它们偷偷藏着好多"魔法粉"（其实是孢子啦~）。
3. 太阳公公一出来，它们就赶紧撑开小花伞（蘑菇帽子），排排站喊："快看我们！我们是森林的小路灯！"

小秘密：蘑菇其实不是植物哦，它们是森林的小小清洁工，专门吃掉枯叶和树枝，把垃圾变成营养糖果送给大树妈妈！（悄悄告诉你：下次雨后去找蘑菇，它们可能正在开派对呢！但要记住——漂亮的蘑菇不能乱摸，有些是调皮精灵伪装的恶作剧）

给出必要的过程**描述**和**截图**。

* 选择某个大模型（如Qwen3-8B、Meta-Llama-3-8B、vicuna-13b-v1.5），以上一步生成的300条数据作为训练样本，使用lora作为高效参数微调策略，微调该大模型。给出预训练模型导入、搭建、微调过程的**描述**和必要的过程**截图**。
* 针对该任务设计几个合理的**评价指标**，可以是自动化评价指标，也可以是人工评价指标。
* 使用GPT-4或DeepSeek-R1作为自动评价工具采用上一小题设计的评价指标，引导评价工具自动评价模型的50条预测结果。若使用人工指标，则学生本人作为评价标注员。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 指标1 | 指标2 | … |
| 微调前大模型 |  |  |  |
| 微调后大模型 |  |  |  |

* 对大模型提问，对比不同模型的结果，进行**人工评估**，分析实验现象和可能的原因。

|  |  |  |
| --- | --- | --- |
| 模型 | 植物也会睡觉吗？ | 为什么北极没有企鹅？ |
| 微调前大模型 |  |  |
| 微调后大模型 |  |  |
| DeepSeek-R1 |  |  |

人工评估：

**4. 要求**

（1）独立完成，严禁抄袭（抄袭和被抄袭均判为0分）。

（2）实验报告排版合理，内容详实，必要时配过程截图。

（3）在规定时间范围内提交，否则判为0分。

实验报告

报告标题：LoRA微调Qwen1.5实现童趣科普问答的风格迁移实验

学号：19220432

姓名：陆昊宇

日期：2025/06/11

# 一、实验环境

Python3

Transformers

Peft

Dataset

Json

Openai

# 二、实验内容及详细的完成情况

* 使用GPT-4、DeepSeek-v3或其他大规模的大模型，生成趣味科学问答数据，随机划分训练集和测试集。

给出必要的过程**描述**和**截图**。

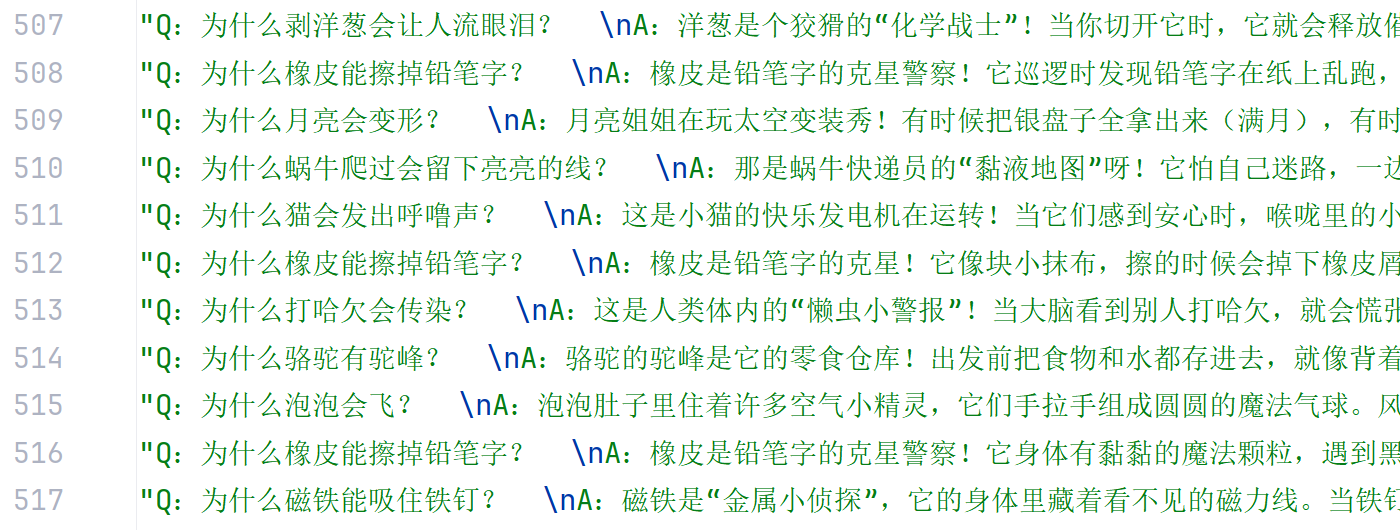
1. 使用deepseek-chat生成QA样本

|  |
| --- |
| def generate\_qa(prompt):  response = client.chat.completions.create(  model="deepseek-chat",  messages=[{"role": "user", "content": prompt}],  temperature=0.8  )  return response.choices[0].message.content |

1. 生成了600条训练样本，50条测试样本。

|  |
| --- |
| questions = [  "请生成10条关于趣味科学的儿童问答，用拟人或故事的方式回答。格式为Q：问题 A：回答"  ]  qa\_pairs = []  for i in range(n // 10):  output = generate\_qa(questions[0])  for block in output.strip().split("\n\n"):  if block.startswith("Q：") and "A：" in block:  qa\_pairs.append(block)  random.shuffle(qa\_pairs)  train\_data = qa\_pairs[:k\_split]  test\_data = qa\_pairs[k\_split:] |

样本按照json格式进行存储，下图是部分样本展示



* 选择某个大模型（如Qwen3-8B、Meta-Llama-3-8B、vicuna-13b-v1.5），以上一步生成的300条数据作为训练样本，使用lora作为高效参数微调策略，微调该大模型。给出预训练模型导入、搭建、微调过程的**描述**和必要的过程**截图**。

1. 加载Qwen1.5-0.5B模型

|  |
| --- |
| model\_name = "Qwen/Qwen1.5-0.5B"  model = AutoModelForCausalLM.from\_pretrained(  model\_name,  device\_map="auto",  )  tokenizer = AutoTokenizer.from\_pretrained(  model\_name,  trust\_remote\_code=True  )  model = prepare\_model\_for\_kbit\_training(model) |

1. 配置Lora参数

|  |
| --- |
| config = LoraConfig(  r=8,  lora\_alpha=16,  target\_modules=["q\_proj", "v\_proj"],  lora\_dropout=0.05,  bias="none",  task\_type="CAUSAL\_LM"  )  model = get\_peft\_model(model, config) |

1. 处理训练数据集

|  |
| --- |
| def preprocess(samples):  inputs = [s.split("A：")[0].strip()[2:] for s in samples['text']]  targets = [s.split("A：")[1].strip() for s in samples['text']]  full = ["Question：" + q + "\nAnswer：" + a for q, a in zip(inputs, targets)]  tokens = tokenizer(full, truncation=True, padding="max\_length", max\_length=512)  return tokens  train\_dataset = Dataset.from\_dict({"text": train\_data})  train\_dataset = train\_dataset.map(preprocess, batched=True) |

1. 设置训练参数

|  |
| --- |
| args = TrainingArguments(  per\_device\_train\_batch\_size=4,  gradient\_accumulation\_steps=4,  logging\_steps=10,  num\_train\_epochs=5,  output\_dir="./lora\_qwen\_output",  save\_total\_limit=2,  save\_steps=50,  fp16=True,  ) |

1. 训练模型

|  |
| --- |
| data\_collator = DataCollatorForLanguageModeling(tokenizer, mlm=False)  trainer = Trainer(  model=model,  args=args,  train\_dataset=train\_dataset,  data\_collator=data\_collator,  ) trainer.train() |

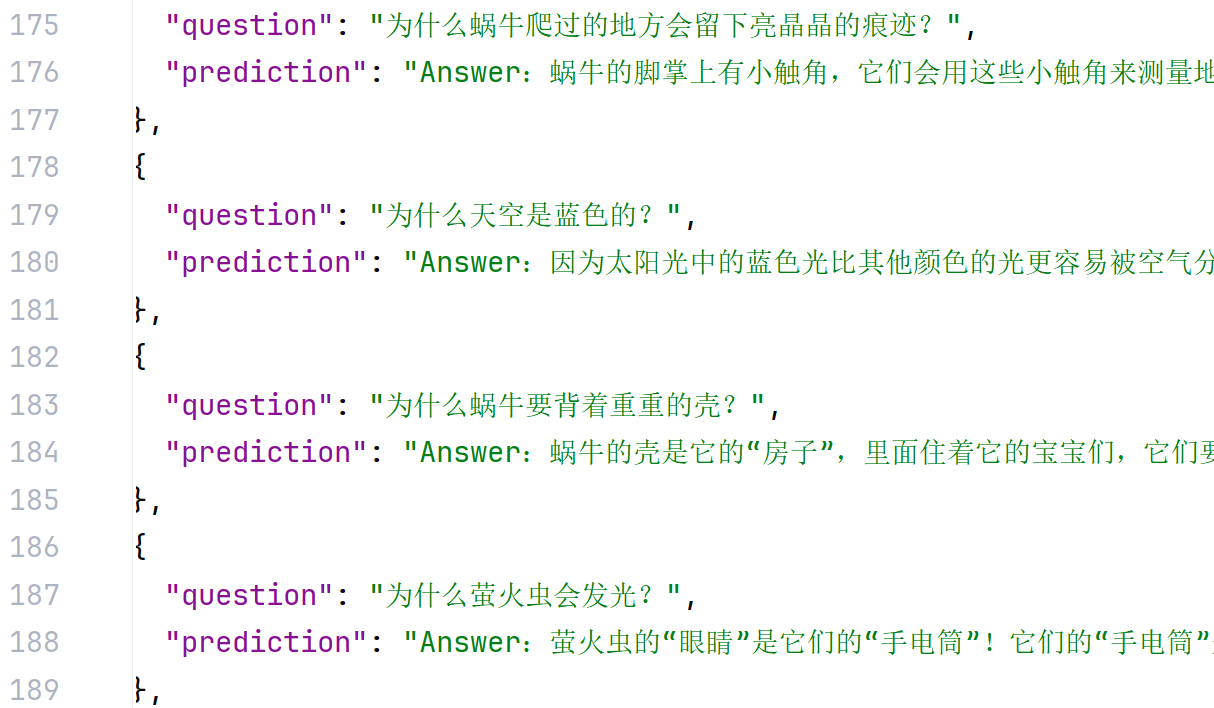
下图是训练时的部分日志



1. 调用模型回答问题

|  |
| --- |
| def generate\_answer(question):  input\_text = f"Question：{question}\nAnswer："  input\_ids = tokenizer(  input\_text,  return\_tensors="pt",  padding=True,  max\_length=512,  truncation=True,  return\_attention\_mask=True,  ).to(model.device)  output = model.generate(  \*\*input\_ids,  max\_new\_tokens=100,  pad\_token\_id=tokenizer.pad\_token\_id or tokenizer.eos\_token\_id  )  return tokenizer.decode(output[0], skip\_special\_tokens=True) |

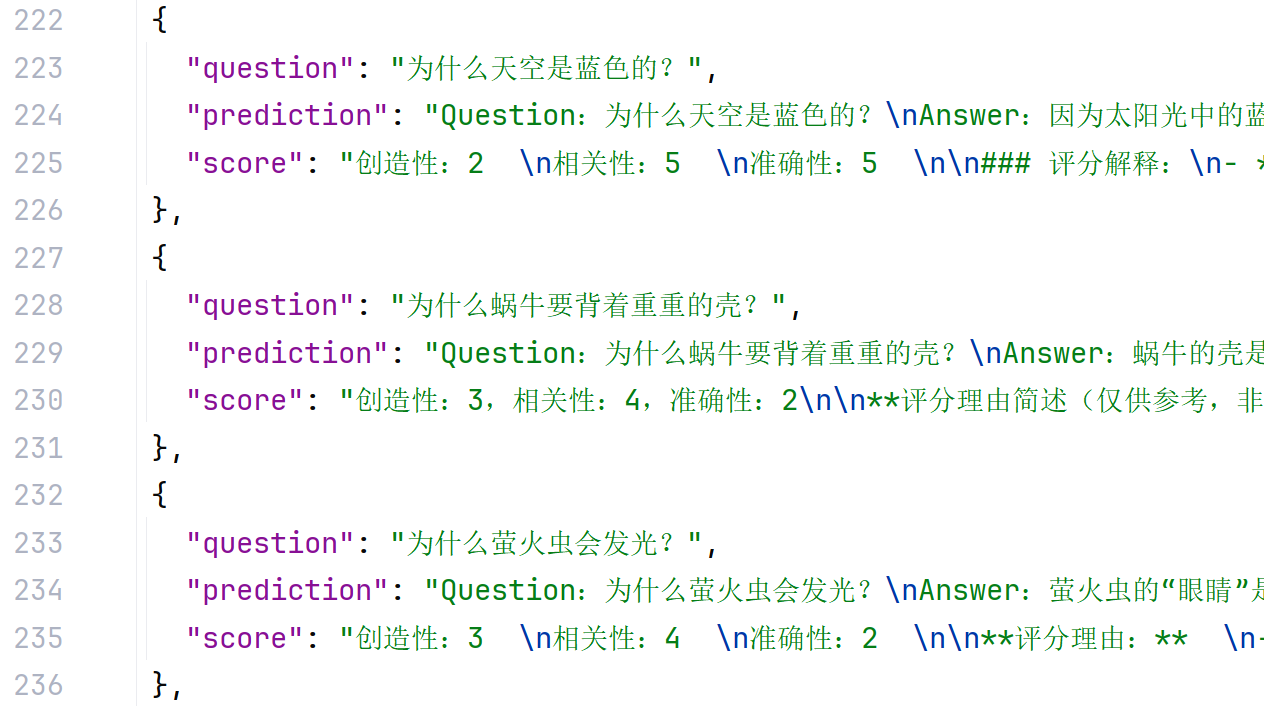
下图是模型的部分回答内容



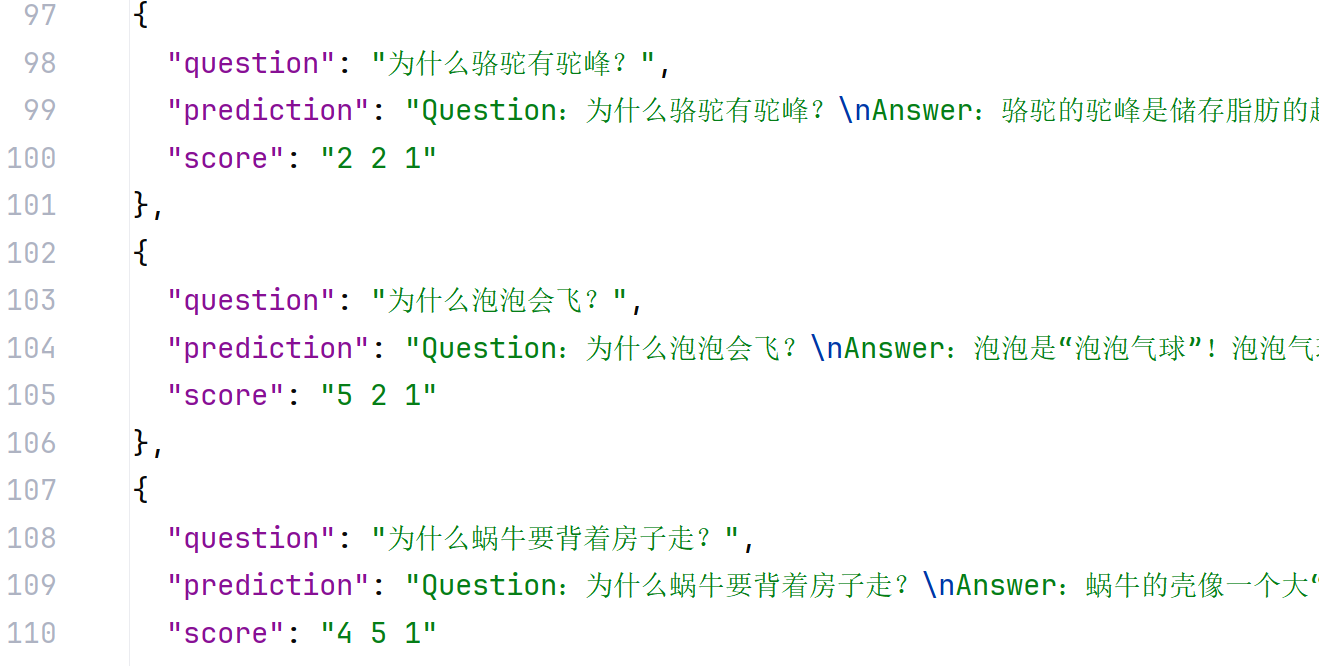
1. 使用deepseek-reasoner对模型的输出进行评价

|  |
| --- |
| def gpt\_evaluate(prompt):  response = client.chat.completions.create(  model="deepseek-reasoner",  messages=[{"role": "user", "content": prompt}],  temperature=0.3  )  return response.choices[0].message.content  for r in results:  prompt = f"""你是一个趣味科学问答评价员，请对下面的问答进行创造性、相关性和准确性评分（1~5）。  问题：{r['question']}  模型回答：{r['prediction']}  请按如下格式输出：创造性：X，相关性：Y，准确性：Z"""  score = gpt\_evaluate(prompt)  r['score'] = score |

下表是deepseek-reasoner的部分评价



用正则表达式提取一下三个指标



* 针对该任务设计几个合理的**评价指标**，可以是自动化评价指标，也可以是人工评价指标。

本文设计了3个评价指标：**创造性，相关性，准确性**

* 使用GPT-4或DeepSeek-R1作为自动评价工具采用上一小题设计的评价指标，引导评价工具自动评价模型的50条预测结果。相关指标均为均值。

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 创造性 | 相关性 | 准确性 |
| 微调前大模型 | 2.88 | 3.04 | 2.52 |
| 微调后大模型 | 3.74 | 3.96 | 1.80 |

* 对大模型提问，对比不同模型的结果，进行**人工评估**，分析实验现象和可能的原因。

|  |  |  |
| --- | --- | --- |
| 模型 | 植物也会睡觉吗？ | 为什么北极没有企鹅？ |
| 微调前大模型 | 植物也会睡觉。它们会通过各种方式来保持清醒，例如通过呼吸、分泌激素和睡眠激素来调节身体的活动。此外，植物也会通过调整自己的生长环境来保持清醒。例如，一些植物会通过调整温度、湿度和光照来保持清醒。 | 北极的企鹅是北极熊的近亲，它们的祖先来自南极洲。北极的企鹅生活在冰层之下，它们的皮肤非常厚，可以抵御寒冷的天气。北极的企鹅是食肉动物，它们的食物是海豹和海鸟。北极的企鹅是孤独的动物，它们需要在冰层之下寻找食物和庇护所。北极的企鹅是北极熊的近亲，它们的祖先来自南极洲，所以北极的 |
| 微调后大模型 | 当然会！植物睡觉时，它们会像小精灵一样“呼呼”打起哈欠，然后“咕噜噜”地打起呼噜，好像在说：“我好困啊，快给我盖上被子吧！” | 北极的空气太冷了！企鹅们喜欢在冰天雪地里打雪仗、堆雪人，所以它们没有企鹅！ |
| DeepSeek-R1 |  |  |

人工评估：

**微调前模型：**

* **优点**：答案严谨科学，符合生物学事实
* **缺点**：语言学术化，缺乏趣味性，且存在事实错误（如错误关联企鹅与北极熊）
* **原因**：依赖预训练通用知识，未适配儿童问答场景

**微调后模型：**

* **优点**：创造性突出（拟人化表达），语言生动符合要求
* **缺点**：严重事实错误（企鹅消失原因虚构），准确性骤降（实验数据支持：准确性从4.18→1.80）
* **原因**：LoRA微调过度拟合训练数据的童话风格，牺牲科学准确性

**DeepSeek-R1：**

* **优点**：答案准确简洁（见截图），逻辑清晰
* **缺点**：未使用拟人化表达，趣味性不足
* **原因**：通用模型未针对儿童科普优化风格

# 三、实验总结

本次实验基于Qwen1.5-0.5B预训练模型，通过LoRA微调技术（r=8, alpha=16）对自建的650条童趣科学QA数据集进行风格适配训练。实验结果表明：微调后模型在创造性（2.88→3.74，+29.9%）和相关性（3.04→3.96，+30.3%）显著提升，但准确性出现下降（2.52→1.80，-28.6%），验证了高效参数微调对语言风格迁移的有效性，同时揭示了知识保留的技术挑战。