ChatGLM 微调改进

卢艳峰

August 12, 2023

前言

- 进一步微调 **ChatGLM2-6B** 模型。
- 阅读 **GPT-1** 和 **GPT-2** 原论文。

stream_chat

```python
def build_stream_inputs(self, tokenizer, query: str, history: List[Tuple[str, str]] = None):
    if history:
        prompt = "\n\n[Round {}]\n\n问：{}\n\n答：".format(len(history) + 1, query)
        input_ids = tokenizer.encode(prompt, add_special_tokens=False)
        input_ids = input_ids[1:]
        inputs = tokenizer.batch_encode_plus([(input_ids, None)], return_tensors="pt", add_special_tokens=False)
    else:
        prompt = "[Round {}]\n\n问：{}\n\n答：".format(len(history) + 1, query)
        inputs = tokenizer([prompt], return_tensors="pt")
    inputs = inputs.to(self.device)
    return inputs
```

stream_chat

```
def format_example(examples): # support question with a single answer or multiple answers
    for i in range(len(examples["prompt"])):
        if examples["prompt"][i] and examples["response"][i]:
            query, answer = examples["prompt"][i], examples["response"][i]
            query = query + examples["query"][i] if examples["query"][i] else query
            history = examples["history"][i] if examples["history"][i] else []
            prompt = ""
            for j, (old_query, response) in enumerate(history):
                prompt += "[Round {}]\n\n问：{}\n\n答：{}\n\n".format(j+1, old_query, response)
            prompt += "[Round {}]\n\n问：{}\n\n答：".format(len(history)+1, query)
            prompt = prefix + prompt
            yield prompt, answer
```

BAAI/bge-large-zh

- 效果并不好。
- 调节了 **VECTOR_SEARCH_TOP_K，SENTENCE_SIZE**
- 测试用例：国家药监局正在进行什么工作？
- 测试用例保存在：《药品标准管理办法》政策解读**.xlsx**
- 目前依旧使用：**GanymedeNil/text2vec-large-chinese**

数据处理：问答数据

- 从 **60** 个 **excel** 文件中读取问答数据，一共 **4707** 条问答数据。

微调效果（无上下文）

- 总共训练 **68 epoch**，最终的 **loss** 为：**0.2588**。大约 **30 epoch** 时 **loss** 开始小于 **1**。
- **batch size**：**4**，**trainable parameters**：**1,949,696**
- 没有收敛，如果继续增加训练时间，**loss** 会继续降低。
- 没有发现明显的灾难性遗忘。
- 如果没有提供上下文，模型依旧是自己瞎编答案。

计划

- 研究常用的评估指标，从定性评估转变为定量评估。
- 阅读 **GLM** 论文，加深对大语言模型的认识。
- 继续简化并重构项目代码，强化核心功能。
- 利用**Sphinx**生成文档，使得项目的生命力更长。

参考

- **ChatGLM-Efficient-Tuning：https://github.com/hiyouga/ChatGLM-Efficient-Tuning**
- **langchain-ChatGLM：https://github.com/imClumsyPanda/langchain-ChatGLM**
- **GPT-1：Improving Language Understanding by Generative Pre-Training**
- **GPT-2：Language Models are Unsupervised Multitask Learners**

Thanks

分享人：卢艳峰