OpenLLM Talk 000

序章

缘起：OpenLLM Talk 这个事情起源于 20230603 OpenLLM 交流群中大家的一场讨论，本着心动不如行动的想法，我们花了一点时间来将其落地，希望可以为大家提供一个 LLM/NLP 领域的**交流平台**。——**我们或许不够 AI，但尽量足够 Open**

注：本期 talk 为 OpenLLM talk 系列的第 0 期，因为本周时间比较紧张，暂时在内容上不做过多要求，以**跑通流程**为主。

整体上分成本周新闻、本周推荐、本周经典（可选）、本周实践（可选）、free talk 等版块，建议后续最好采用每个版块每期由 1-2 人认领+多人参与贡献+自由讨论的形式。

【编号】：OpenLLM Talk 000 (三位数是希望 LLM 的热度+我们的热情+读者的热情可以支撑我们做到三位数)

【时间】：20230610 晚上十点

【本期提要】：本期以跑通流程为主，内容上可以看看本周经典和 free talk

【本期贡献者】- 排名不分先后：

【主持人】：羡鱼（后续每期由大家自行认领）

【版块负责人】：羡鱼（后续每期由大家自行认领）

【具体内容贡献者】：请查看具体内容后面的署名，比如问题、回答和观点的来源

本周新闻

【本周新闻】：LLM/AI news，包括但不限于学术、项目、工业界新闻和进展；多人认领或者直接在此添加，由 **1-2 人认领并汇总**；建议大家都参与进来，相互补充，尽量**减少信息冗余和缺漏**；共 ~10 分钟；

【贡献者】：

【建议区】：可以考虑 GitHub 的讨论区，看个人习惯；

学术

项目

工业界

2023 智源大会

https://2023.baai.ac.cn/

Chatgpt 与科学计算专题论坛

https://play.itdks.com/watch/10982282?player=

本周推荐

【本周推荐】：本周重点内容推荐和介绍，模型、开源项目、好的资料或课程，建议 1-3 项；共 15 分钟；

【贡献者】：

【提名区】：**Openbuddy**；

【建议区】：固定收集部分评测 case；跑通 langchain 的功能；

模型

项目

其他

本周经典-optional

【本周经典】：NLP/LLM 领域的经典话题探讨；~15 分钟；

【贡献者】：羡鱼

【提名区】：位置编码

【本周主题】：位置编码

【OpenLLM 009】大模型基础组件之位置编码-万字长文全面解读 LLM 中的位置编码与长度外推性（上） - 羡鱼智能的文章 - 知乎

https://zhuanlan.zhihu.com/p/626828066

本周实践-optional

【本周实践】：NLP/LLM 领域实践经验分享，可以分享自己的实践经验或者他人的实践经验，后面群里也会组织一些实践内容；~15 分钟；

【贡献者】：

【提名区】：

Free Talk

【Free Talk】自由提问，自由讨论；在文档里提问或者在群里提问，建议尽量在此汇总；如果群里已经有比较好的讨论结果，也可以将讨论结果搬运过来；时间不限；

【贡献者】：羡鱼（编辑）+OpenLLM 群友

Q1：falcon 相比 llama 有何特殊之处？—羡鱼

A1：

Q2：掩码的左右问题，为啥分左右，区别是啥？—群友

Q3：比较想了解一下大模型的细节，我看里面有 attention mask 啥的，还有 tokenizer 和

transformer 具体分别在干什么事情？tokenizer 是将每个词转成了向量吗？text embedding 的工作原理是什么？—yuhan

A3：

推荐阅读：

哈佛 **注释版 transformer**

1）attention mask：屏蔽掉不想要 attend 的东西， 具体取决于你的使用场景，比如 padding 项、未来的 token、一些复杂模型的 attention 机制（比如 GLM 里面的 attention）；-from 羡鱼

2）transformer：

3）tokenizer：Google 的 **sentencepiece** 工具，主流的 **BPE、ULM**

https://github.com/google/sentencepiece.git

4）text embedding：**一般叫 sentence embedding**，做法很多，比如 avg/sum/cls/last token。

fastchat 的实现-avg：--群友

```
    def get_embedding(input):
        """Get embedding main function"""
        with torch.no_grad():
            encoding = tokenizer.batch_encode_plus(
                input, padding=True, return_tensors="pt"
            )
            input_ids = encoding["input_ids"].to('cuda')
            attention_mask = encoding["attention_mask"].to('cuda')
            model_output = model(
                input_ids, attention_mask, output_hidden_states=True
            )
            data = model_output.hidden_states[-1]
            mask = attention_mask.unsqueeze(-
    1).expand(data.size()).float()
            masked_embeddings = data * mask
            sum_embeddings = torch.sum(masked_embeddings, dim=1)
            seq_length = torch.sum(mask, dim=1)
            embedding = sum_embeddings / seq_length
            normalized_embeddings = F.normalize(embedding, p=2,
    dim=1)
            ret = normalized_embeddings.tolist()
        return ret
```

那么 openai 的 embedding 接口是咋实现的，如何由 token 的表示得到句子的表示？-yuhan

Q4: reward model 哪家好？ -qingwang

A4：个人感觉 reward model 和具体的数据比较相关-羡鱼；ChatGPT 做通用的打分应该还可以；

Q5: 千亿模型的训练成本分析 -qingwang

A5：300B token 175B 模型 **1000 A100 卡** 最少**一个月**

Q6: 小于 8bit 的量化在推理的时候是怎么计算的？此时 tensor 的 data 指针应该设置成什么值？-Dark Flame Master

A6：**个人猜测，指针会对齐到 8bit**，但是 GPU 计算可以一次算 N 个 4bit 数据 即 packed。-qingwang

5 Levels Of Summarization ：5 种使用 langchain summarization 的层次

时间关系-本期就先到这儿，欢迎关注我们之后的更多精彩内容！

Q11: 现在语言模型中有使用用 LoRA 作为风格的案例吗？目前了解到的 LoRA 在语言模型中主要是用于优化。

A：

参考资料

2023 智源大会

https://2023.baai.ac.cn/

Chatgpt 与科学计算专题论坛

https://play.itdks.com/watch/10982282?player=

【OpenLLM 009】大模型基础组件之位置编码-万字长文全面解读 LLM 中的位置编码与长度外推性（上） - 羡鱼智能的文章 - 知乎

https://zhuanlan.zhihu.com/p/626828066

哈佛 **注释版 transformer**