LLM 全景图 （The Landscape of LLM）

Original 安迪的写作间 [安迪的写作间](javascript:void(0);) 2023-06-14 17:08

前言：最近 LLM 大火，相关领域熟人都想往这个方向靠，好几个朋友咨询了这块，也有帮做模拟面试。最近有个哥们让我推荐对这个领域整体快速了解的资料。于是我就花了些时间整理资料，希望可以俯瞰当前 LLM 领域，能够给个整体的印象，如若对某个细分板块感兴趣，读者也可以找更深入的资料。

本文假设读者有一定相关领域的知识，至少是机器学习，深度学习，或者数据科学的经验。

公众号回复 llm，获取 pdf 资料包，各个资料链接可以点看原文看知乎的文章（没出来就在审核）。



**Overview 概观**

* 首先 Andrej 的 State of GPT 是必看的：对整个 LLM 的全景讲的很清晰，不亏是我辈 CS231n 的好老师，高屋建瓴
* W&B 关于 LLM 的 Tutorial 干货不少：Current Best Practices for Training LLMs from Scratch
* 两篇比较好的综述：
  + An Overview on Language Models: Recent Developments and Outlook：偏训练技巧
  + Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond：偏应用

**前沿模型观摩**

* GPT4 Technical Report：GPT4技术报告
  + Spark of AGI[视频]：对 GPT4 的详细测评
* PALM2 Technical Report：PALM2技术报告

大概看看现在最好能做到什么程度，虽然细节干货少。

接着按不同模块分别推荐些我能想到的比较重要入门论文和资料，精力有限，肯定是挂一漏万，欢迎指正完善。

**预训练**

**整体**

预训练可先看一些经典基座模型训练流程：

* GPT3 论文：不用多说
* LLAMA：经历各种检验最靠谱英文模型，很多训练细节值得参考
* BloombergGPT：虽是金融领域模型，但各种思想都类似，很踏实，特别给训领域模型借鉴
* Transformer Math 101：很干的一篇关于训练中各种计算的博客，扎实
* [可选] 谷歌系的：PaLM，Flan-PaLM：一个预训练，一个指令微调，大公司还是挺多insight可学习
* [可选] BLOOM，BLOOMZ：可以大概看看，比较糙，特别是和大公司几篇对比

**数据相关**

* W&B 关于预训练数据处理较全面博客：Processing Data for Large Language Models
* 实际数据集参考，会有各种细节：
  + ROOTS：BigScience 开源数据集
  + The RefinedWeb Dataset：最近模型 Falcon 的数据集
  + [可选] RedPajama：号称复刻 LLAMA，没有报告，更多都在源代码里面，可以简单看看有时间看代码
  + [可选] The Stack：代码领域的数据集，感兴趣也值得一看
* 近期一些关于数据推荐论文：
  + Scaling Data-Constrained Language Models
  + A Pretrainer's Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity

**训练相关**

* Megatron-DeepSpeed：现在基本都基于这套框架训，有必要浏览下原论文
* DeepSpeed 相关：ZeRO优化器必看，可能看视频会更好理解些
* 混合精度及其他相关训练技巧：
  + Performance and Scalability:How To Fit a Bigger Model and Train It Faster Hugging face相关博客
  + 更多关于混合精度的讨论，Nvidia 相关文档，看2和3就行，关于 BF16 的讨论（结论能上BF16就上）
* lilian写的关于大规模分布式训练相关的综述：How to Train Really Large Models on Many GPUs?

**Tokenization**

* Huggingface 的 Tokenizers 的相关 Tutorial：看下面这几节就行
  + Normalization and pre-tokenization
  + Byte-Pair Encoding tokenization
  + [感兴趣看] WordPiece tokenization
  + [感兴趣看] Unigram tokenization
  + Building a tokenizer, block by block
* [可选]  SentencePiece 的 Readme，就是另一个训练词表的库

**SFT(Supervised FineTuning)/Instruction Tuning**

就是搜集到指令数据，然后直接 finetune 训练，难点在于在于怎么收集到较好的SFT数据。

主要可分成下面几种方法：

* 人工标注写
* 基于模板
* 基于开源共享数据
* 网上相似数据爬取
* OpenAI API接口通过 Prompt 工程构建

**SFT 模型**

* 基于模板：
  + FLAN 模型：前 ChatGPT 时代，可以看看如何基于模板构建一些传统指令数据，还出了v2
* 基于 OpenAI API 和开源共享数据（后 ChatGPT 时代）：
  + Exploring the Impact of Instruction Data Scaling on Large Language Models
  + Towards Better Instruction Following Language Models for Chinese
  + 直接从 ChatGPT 和 GPT4 抓数据来训SFT：Alpaca，Vicuna，GPT4ALL
  + 中文这块：Belle 的两篇论文也可以简单看看，看怎么构建数据（比如Self-Instruct）
* 基于网络爬取数据及人工标注
  + LIMA: Less Is More for Alignment：怎么构建一个多样化的SFT数据集

**RLHF 部分**

建议给 OpenAI 在这块的努力都刷一遍，会有很清晰了解，各种细节。现在大多引 22 年 InstructGPT，但 OpenAI 这整套流程也都是一步步完善出来的。最早期在 GPT2 就探索用 Human Preference，中间如何尝试、遇到什么问题、以及怎么解决都写得挺清楚，甚至还专门写了怎么培训标注人员，当时 OpenAI 是真的挺 Open：

* Fine-Tuning GPT-2 from Human Preference（Blog）：做 SFT 来获得人类偏好，但 SFT 数据都是模型采样后人标出来的
* Learning to summarize from human feedback（Blog）：基本这套 SFT、Reward Model、RLHF 流程齐全了，用在 Summarization 任务，只标了 Pair-wise 的数据，而不是后面的 Rank
* Summarizing Books with Human Feedback（Blog）：当摸清楚这套流程之后，OpenAI 开始对 RLHF 增大规模，也开始发现生成这块的评估难，引入模型辅助的评估
* WebGPT: Improving the Factual Accuracy of Language Models through Web Browsing[Blog]：OpenAI 想用强化来训练模型浏览网络，然后做检索增强
* InstructGPT：Training language models to follow instructions with human feedback[Blog]：最后才到 InstructGPT，罗马不是一天建成，能看到 OpenAI 在 RLHF 上是积累了很长时间的，到 InstructGPT 可能是 API 用户反馈有类似需求，当时没引起太大反响，直到加入了对话数据，并且给模型放出来给大家用，也就是 ChatGPT.

**推理使用**

**解码方法**

* 关于 Search 和 Sample，还有 Topp Topk 讨论：献丑了

**Prompt Engineering**

Zero-shot 和 Few-Shot 给 GPT3 那篇看完就差不多懂了，这里列些较经典方法

* Chain-of-Thought Prompting Elicits Reasoning in Large Language Models：思维链，开山之作，本质上给LLM模型更多 token 来用来思考，增加结果准确性
* Self-Consistency Improves Chain of Thought Reasoning in Language Models：思维链后续，其实就是类似search算法给搜索空间扩大，比如beam从1改成多个，然后最后结果 ensemble 一下
* Tree of Thoughts: Deliberate Problem Solving with Large Language Models：给链扩展成树，进行检索和集成，上面两种都变成树的某个特例
* [可选] ChatGPT Prompt Engineering for Developers：吴教授 DeepLearning AI 关于 ChatGPT prompt 相关

**反思以及其他高阶应用**

* Reflexion: Language Agents with Verbal Reinforcement Learning：提出反思系统，模型能对自己之前的结果进行反思，之后再做出修改
* AutoGPT介绍：基于 GPT API 非常酷的应用，通过设计维护几个GPT角色，还有Memory系统，给AutoGPT一个目标和初始任务，就能让它成为一个无情的做任务机器人，完成任务，创造新任务，排好优先级，继续做任务
* [可选] Generative Agents: Interactive Simulacra of Human Behavior：写 AutoGPT 的时候莫名让我想起这篇，虽然是讲用GPT做游戏任务的思考机器，但是本身里面就涉及到大量应用相关的设计
* [可选] Large Language Models as Tool Makers：idea 也很酷，维护几个GPT角色，让一个做工具，一个使用工具解决问题，还有个判断什么时候需要做新工具，也是系统的设计

**其他**

**Tools （API 调用）**

有些能力调外部接口会容易很多，包括最新信息获取

* Toolformer: Language Models Can Teach Themselves to Use Tools：占坑之作，方法较 naive
* TaskMatrix.AI: Completing Tasks by Connecting Foundation Models with Millions of APIs：给 Tools 理念推到更大的框架
  + [可选]对这个领域兴趣，还可以看Gorilla，HuggingGPT
* 搜索增强LLM（虽然我认为检索也只是一个工具）：REPLUG: Retrieval-Augmented Black-Box Language Models

**Code Related**

* CodeX：Evaluating Large Language Models Trained on Code：Code相关必读，现在普遍认为Code对推理能力会有一定提升

**Math Related**

因为数学涉及到模型的推理能力，所以一般认为较重要

* Training Verifiers to Solve Math Word Problems：让模型解数学题，提出了 Verifier 的思路，就是对多条打分选最高。

**[可选] GPT API相关应用**

* Building Systems with the ChatGPT API：吴教授关于 OpenAI API应用的课程
* [可选] OpenAI CookBook：OpenAI 官方的 API 使用说明，里面很多特别实用的建议

**[可选] 损失函数**

* Efficient Training of Language Models to Fill in the Middle：无损 FIM 损失，增加模型中间填充能力
* UL2: Unifying Language Learning Paradigms：对T5的损失进行改进，兼容考虑了GPT单向生成损失，PALM2声称用的是它的变种