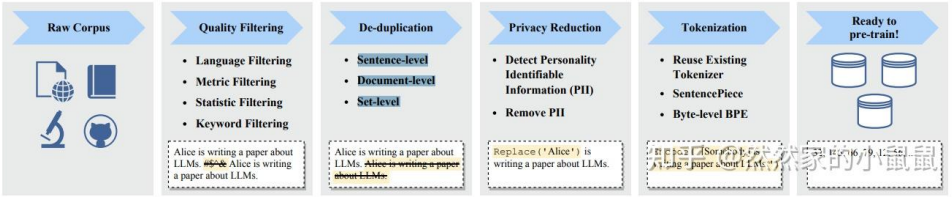
数据清洗

pretrain

- 预训练数据

表 4.1: 预训练数据集概览

数据集	发布者	规模	特点	支撑的语言模型
BooksCorpus	Zhu et al. (2015)、Shawn Presser	book1: 2.2GB; book3: 37GB	英文小说集	GPT 系列、OPT、OPT-IML
Wikipedia	维基媒体基金会	21.23 GB	多语言高质量百科全书	GPT 系列、OPT、OPT-IML
Common Crawl	Common Crawl 团队	超过 PB	网页数据，规模巨大	GPT 系列、T5、UL2、Flan-T5
ROOT	BigScience	1.6TB	包含 69 种语言	BLOOM、BLOOMZ、mT0
The Pile	Gao et al.(2020)	825GB	数据来源广泛，多样性佳	GLM-130B、GPT-J、GPT-NeoX-20B、OPT、OPT-IML、GLM-130B
悟道	北京智源人工智能研究院	3TB	中文数据集	GLM-130B
CLUECorpus 2020	GLUE 开源社区	100GB	中文数据集	
MNBVC MNBVC	里屋社区	2.18TB	中文数据集	

- 清洗预训练数据流程
  - 清洗一般步骤：质量过滤–去重–隐私删除–Tokenization
    - 

    - 质量过滤
      - 基于分类器
        - 训练一个二分类器，判断高质量与低质量文本
      - 基于启发式
        - 基于语言过滤（去除不相关的语言）
        - 基于指标过滤，例如困惑度等
        - 基于统计过滤，使用标点分布、符号词比、句子长度等过滤低质量
        - 基于关键词过滤：过滤包含关键词的文本、例如链接、引用等
    - 去除重复文本
      - 句子级别、文件级别、词集级别
      - Deduplicating Training Data Makes Language Models Better
        - paper：https://arxiv.org/pdf/2107.06499v1.pdf
        - Deduplicating Training Data Mitigates Privacy Risks in Language Models
          - paper：http://proceedings.mlr.press/v162/kandpal22a/kandpal22a.pdf
        - 结论：去重可以加快模型训练速度、不易受到隐私攻击
    - 去除文本中的隐私信息
      - 住址、电话等
    - Tokenization

        - SentencePiece
        - BPE
        - https://zhuanlan.zhihu.com/p/630696264
    - When Less is More: Investigating Data Pruning for Pretraining LLMs at Scale
        - paper：https://arxiv.org/pdf/2309.04564.pdf
        - 步骤：语言识别（剔除无关数据）–规则方法–轻量级模型过滤–数据去重
            - 语言识别，删除与目标语言无关的数据
            - 规则方法：
                - 丢弃perplexity高的文本数据（过滤掉不自然的文本）
                - 删除标点/符号过多、过长/过短的句子
                - 删除具有某些特定词汇（如链接、脏话、敏感词）的句子
            - 通过轻量级模型进行过滤，例如KenLM，避免噪音
                - KenLM
                    - code:https://github.com/kpu/kenlm
                    - 使用：https://www.zhihu.com/tardis/bd/art/399494766
            - 数据去除重复/相似的信息，在文档层面进行模糊重复数据删除，例如通过minHash来删除相似文档
                - 包含大量重复词汇或短语的句子可以删除
                - 重复率（词/n-grams共现）过高的段落删除
                - 删除训练集中与测试集相关度过高的内容
        - 结论：
            - 基于困惑度对50%/30%的数据集剪枝比使用错误L2范数效果好
            - 删除简单实例可以提高模型性能，简单实例比如困惑度低的底层样本，错误L2范数指标低的底层样本、使用记忆话指标的顶部样本
        - 开源工具：
            - FastText：https://github.com/topics/fasttext?o=asc&s=forks
            - CC–Net：https://github.com/facebookresearch/cc_net
            - MinHashLSH：https://github.com/topics/minhash–lsh–algorithm

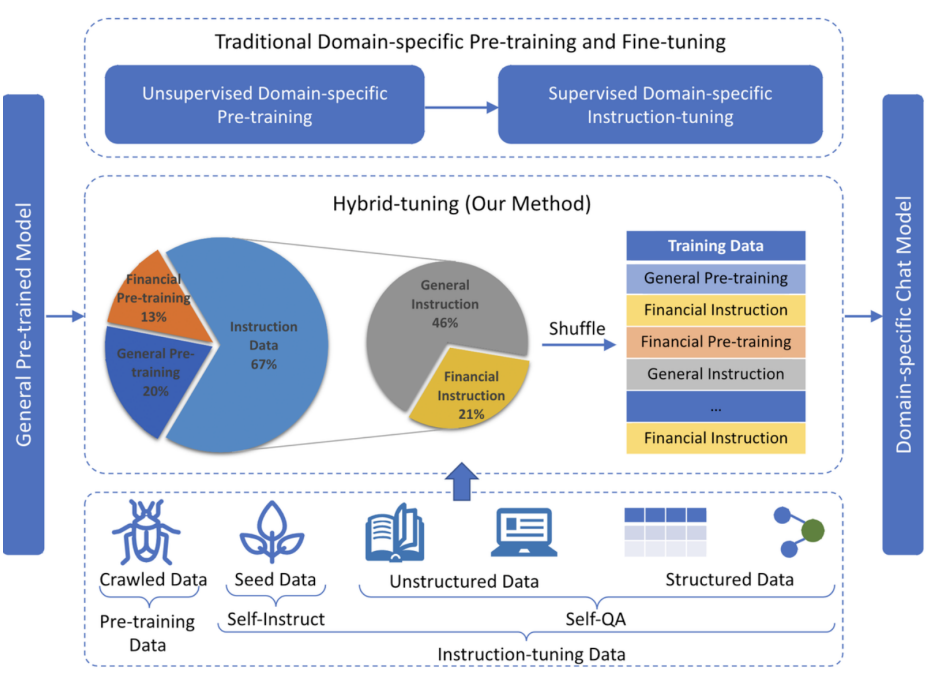
算法 / 步骤	KSentence	KShingle	MinHash	SimHash
清洗文本	去掉部分标点和特殊字符，保留逗号、句号、冒号、换行等语句分隔标点	去掉标点，空格，特殊字符等	去掉标点，空格，特殊字符等	去掉标点，空格，特殊字符等
分割词组	无	指定长度的滑动窗口提取Shingle作为词组	指定长度的滑动窗口提取Shingle作为词组	分词结果作为词组
分割语句	根据语句分隔标点提取文本语句	无	无	无
生成词库	无	所有文本的互异词组的集合构成词库	所有文本的互异词组的集合构成词库	无
提取指纹	计算语句的长度，提取K个最长的语句，拼接文本后计算MD5作为指纹	基于词库计算文本的one-hot编码向量作为文本指纹	基于词库计算文本的one-hot编码，再对文本的one-hot编码向量随机排列，取首个非零元素的下标，重复N次，得到的N维向量作为文本指纹	统计文本中各词组的词频，将各词组哈希为指定长度的01向量，将0改为-1，按位与对应词频相乘，再将所有词组的相乘结果按位相加，大于0映射为1，否则映射为0，得到的01向量作为文本指纹
建立索引	以指纹为索引，指向具有该指纹的文本ID	以指纹为索引，指向具有该指纹的文本ID	将文本指纹分段，以各段指纹为索引，指向具有该指纹的文本ID	将文本指纹分段，以各段指纹为索引，指向具有该指纹的文本ID
生成候选	具有同一指纹的文本对	具有同一指纹的文本对	至少具有同一段相同指纹的文本对作为候选	至少具有同一段相同指纹的文本对作为候选
计算距离	无需计算，默认所有候选文本对都是重复文本对	计算候选文本对的Jaccard相似度，大于阈值则认为是重复文本对	计算候选文本对的同位元素相等概率，大于阈值则认为是重复文本对	计算候选文本对的Hamming距离，小于阈值则认为是重复文本对

        - 案例：CulturaX数据集清洗
            - paper：https://arxiv.org/pdf/2309.09400.pdf
            - code：https://huggingface.co/datasets/uonlp/CulturaX
- 评估预训练数据质量
    - 困惑度
    - 错误L2范数
    - 记忆化

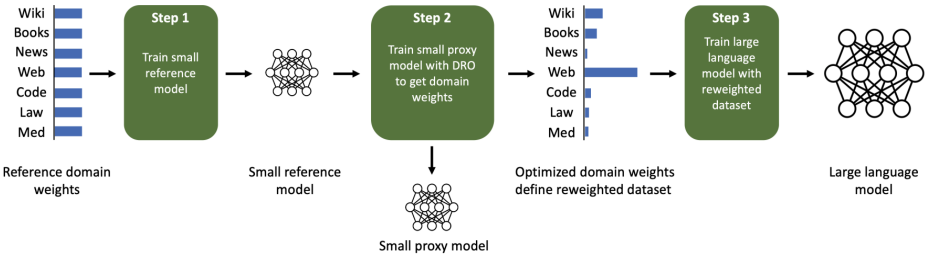
sft

- 构造数据
  - 《SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions》
    - paper：https://arxiv.org/pdf/2212.10560.pdf
    - code：https://github.com/yizhongw/self-instruct
    - 步骤：指令生成-实例生成-过滤&后处理
      - 指令生成
        - 每次从任务池中抽取8条数据（一个任务=一条指令+一条实例），注意任务/场景粒度是有限的
      - 实例生成
        - 将指令给LLM生成对应的数据
      - 过滤&后处理
        - 衡量新数据与旧数据的相似度，只有当ROUGE-L相似度小于0.7的时候才可
- 清洗数据
  - 数据污染问题
    - 输入与输出污染
      - 预训练数据中存在与下游任务标签相同的数据，模型倾向于复制文本，而非学习真正的解题模式
    - 输入污染
      - 评估样本中并未包含标签情况，导致下游任务的性能高估。在做零样本和少样本评估的时候，如果预训练数据集中存在与热门基准任务重叠的数据，我们必须重视数据去污染。
  - 消除错误&不一致
  - 项目
    - https://github.com/gururise/AlpacaDataCleaned
      - 清洗代码：https://github.com/gururise/AlpacaDataCleaned/tree/main/tools
        - 检查数据集的input是否存在潜在问题
        - 询问LLM给定的指令与输入是否正确、对应
        - 检查数据集中是否有合并的指令
        - 检查output是否有问题，是否有截断现象、输出是否符合字典格式等
- 筛选数据
  - From Quantity to Quality: Boosting LLM Performance with Self-Guided Data Selection for Instruction Tuning 数学公式
    - paper:https://arxiv.org/abs/2308.12032
    - code:https://github.com/tianyi-lab/Cherry_LLM
    - 核心：提出一个指令跟随难度指标IFD，通过该指标来筛选具有增强LLM指令调优潜力的数据样例
    - IFD计算公式：
      - $$CAS = s_\theta(A|Q) = \frac{1}{N}\sum_{i=1}^{N}\log p(w_i^A|Q,\ldots,w_{i-1}^A;\theta)$$
        - CAS表示条件回答分数，用于表示模型对指令Q生成答案A的难易程度
      - $$DAS = s_\theta(A) = \frac{1}{N}\sum_{i=1}^{N}\log P(w_1^A,\ldots,w_{i-1}^A;\theta)$$
        - DAS表示直接答案分数，是answer部分的损失
      - $$IFD = r_\theta(Q,A) = \frac{s_\theta(A|Q)}{s_\theta(A)}$$

        - 指令跟随难度，数值越大对模型越有利
    - 步骤：少量数据微调&使用聚类来保证数据多样性-微调模型计算IFD-重新训练
        - 先少量数据1k用于模型sft微调，训练出v1版模型，需要保证数据多样性
        - 使用该微调模型计算所有数据的IFD，筛选数据
        - 使用筛选的数据重新训练模型
    - 结论：模型仅仅使用5%-10%的数据就可以达到全量数据的效果，甚至有提升
    - 启发：可以通过计算数据的IFD，自动筛选top k的数据用于微调，另外可以借鉴基于困惑度+聚类的思想来保证数据的多样性
- LIMA: Less Is More for Alignment
    - paper：https://arxiv.org/pdf/2305.11206.pdf
    - 假设：模型在预训练见过所有数据，sft只是学习交互方式和风格
    - 描述：65B的微调LLM，没有使用rlhf
    - 核心实验：
        - 单论对话：使用1k的高质量sft数据，人工手写250条+社区问答750，并且在prompt中加加入一步一步思考（Let's think step by step）
        - 多轮对话：30个高质量对话样本
    - 结论：少量高质量的数据可以大幅提升模型对应的能力
- Exploring the Impact of Instruction Data Scaling on Large Language Models: An Empirical Study on Real-World Use Cases
    - paper:https://arxiv.org/pdf/2303.14742.pdf
    - code:https://github.com/LianjiaTech/BELLE
    - 结论：
        - 对于翻译、改写、头脑风暴任务，200w甚至更少的数据可以使得模型效果更好
        - 对于提取、分类、封闭式QA和总结摘要任务，效果随着数据量的提升而提升
        - 模型在数学、代码、CoT上表现差，可以尝试在数据质量、模型规模、训练策略上进行改进
- Duxiaoman-DI
    - code:https://github.com/Duxiaoman-DI/XuanYuan
    - 核心：基于Bloom176B，使用混合微调缓解灾难性遗忘
    - 混合数据微调步骤
        - 从网上爬取预训练数据+清洗过滤
        - 使用self-instruction收集通用数据，构造为结构化和非结构化数据
        - 按照一定比例进行混合一般指令数据：金融数据



- DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining
  - paper：https://arxiv.org/pdf/2305.10429v2.pdf
  - code：https://github.com/sangmichaelxie/doremi
  - 核心：提出基于域权重的算法，使用域上的组分布鲁棒优化训练一个小型代理模型，用于生成域权重，然后根据域权重新训练一个更大的模型



- Instruction Mining: High-Quality Instruction Data Selection for Large Language Models
  - paper：https://arxiv.org/pdf/2307.06290v1.pdf
  - 核心：一种线性质量规则和评估指标
    - 已有的指标：length/rewardscore/perplexity/MTLD/KNN-i/unieval-naturalness/coherence/understandability
  - 结论：
    - PPL、MTLD、Nat、Und与预期评估损失呈现正相关，越低越好
    - rewardscore、coh与评估损失呈现负相关，越高越好

工具

- 去除重复数据工具
  - deduplicate-text-datasets
    - code：https://github.com/google-research/deduplicate-text-datasets
  - datasketch
    - code：https://github.com/ekzhu/datasketch
  - data-juicer
    - paper:Data-Juicer: A One-Stop Data Processing System for Large Language Models (arxiv.org)

        - code:https://github.com/modelscope/data-juicer/releases/tag/v0.2.0
        - 服务器代码：/mnt/pfs/jinfeng_team/SFT/qinxiuyuan/workplace/dataprocess/data-juicer
        - 环境：conda activate dataJuicer
        - 三种去重模式
            - md5
            - MinHash
                - tokenizer
                - 窗口级别的向量计算Jaccard系数
            - SimHash
                - 分词-Hash-加权-合并-降维
                    - 分词（启发式、jieba分词等）
                    - Hash，利用hash函数计算每个特征向量的hash值
                    - 加权：将所有分词得到的hash值与其对应的权重进行加权求和
                    - 降维：将原先的数值降低到只使用0-1表示
    - 清洗污染数据工具
        - lm-eval harness
            - code：https://github.com/EleutherAI/lm-evaluation-harness
            - 用法：https://zhuanlan.zhihu.com/p/671235487
    - 微调数据处理工具
        - code：https://github.com/modelscope/data-juicer/releases/tag/v0.2.0
- Dataverse: Open-Source ETL (Extract, Transform, Load) Pipeline for Large Language Models
    - paper：https://arxiv.org/pdf/2403.19340.pdf
    - code：https://github.com/UpstageAI/dataverse
    - 支持数据去重、数据清洗、PII个人身份信息移除、数据质量提升、消除偏见、去除毒性数据等功能

参考资料

- CSDN博客
    - sft和pretrain数据处理和筛选方法：https://blog.csdn.net/qq_35812205/article/details/134104120
    - minHash计算：https://blog.csdn.net/zfhsfdhdfajhsr/article/details/128529402
- 知乎
    - 数据中心：https://zhuanlan.zhihu.com/p/617057227
    - 《A survey of Large Language Models》：https://zhuanlan.zhihu.com/p/631065995
- Github
    - instructionZoo:https://github.com/FreedomIntelligence/InstructionZoo
    - data-centric-AI：https://github.com/daochenzha/data-centric-AI
- 其他：
    - SimHash计算：https://blog.51cto.com/u_16099328/9147103
    - LLM数据处理：https://wandb.ai/wandb_gen/llm-data-processing/reports/Processing-Data-for-Large-Language-Models--VmlldzozMDg4MTM2