FT-Data-Ranker 7B 赛道技术报告

DiMiner 团队 (ID: 946288)

摘 要 本技术报告为 DiMiner 团队 (ID: 946288) 针对 FT-Data-Ranker 7B 赛道的具体技术与实施细节说明。FT-Data-Ranker 7B 赛道关注大型语言模型的 fine-tuning 数据的筛选，需要对指定原始数据集进行清洗、过滤和增强，然后利用新数据集按照指定流程对指定大型语言模型进行微调，提高测试集上的效果。本方案在该赛道原本提供的数据筛选算子的基础上，引入了当下最新的数据筛选思路，通过额外训练一个学习了更大数据量的指导模型来对最终训练的数据进行熵计算和筛选。该方案在两阶段的比赛中均取得了较好的成绩。

关键词 FT-Data-Ranker 7B 赛道；数据筛选；大模型微调

1 引言

FT-Data-Ranker 7B 赛道* 是针对大型语言模型 (LLM) 的微调 (fine-tuning) 数据自动化的进行清洗、过滤和增强，并探究如何得到更好的微调的比赛。其原始数据来源于 Alpaca-cot 数据†，并要求在 baichuan2-7B 这一 LLM 上[1]进行固定设置的 LORA 微调。

为了实现该比赛中自动化筛选数据以及得到更好的微调结果的要求，本团队构建了基于 LLM 的熵计算进行自动数据筛选的方案。该方案的思想在于在同样的训练的条件下，先使用更多的数据去训练一个的模型作为指导模型，用原始模型和该指导模型分别去计算每一条数据的在训练前后的熵 (entropy)；接着再基于 LoBaSS 论文的假设[2]，只有在指导模型上 entropy 比原始 entropy 小的数据才是模型在该模型在该条件下能学习到的数据，并以此为标准进行数据筛选；最终，从这些 entropy 变小的数据中按比赛要求中英文 1：1 抽样 10M 的 token，进行最终模型的训练。通过我们的测试发现，该方法不仅能在中英文 1：1 抽样得到更好的效果，在更多的数据比例的实验下，也能取得有效的提升。

2 技术方案

该技术方案的核心在于，基于 LoBaSS 论文的假设[2]，我们认为在同一个模型训练前后，只有在该数据 entropy 值减小的情况下，该数据是能被模型学习到的，否则该数据并并不能通过模型学习得到改善，应该被剔除。因此，本技术方案将先训练一个指导模型并和原模型分别计算数据的 entropy 进行筛选。本技术方案将包含如下四个部分：初始数据采样，指导模型训练，数据筛选和最终模型训练。

*https://tianchi.aliyun.com/competition/entrance/532158

†https://github.com/PhoebusSi/Alpaca-CoT

2.1 初始数据采样

基于原提交指南 **3.1 改良原始数据集**得到的 en_refine.jsonl 和 zh_refine.jsonl 数据，采用原提交指南 **3.2 数据集采样**的 get_train_dataset_7b.py 工具进行多次采样备用：

1. 10m_en_refine.jsonl：仅含英文的 10m token 的数据集
2. 10m_zh_refine.jsonl：仅含中文的 10m token 的数据集
3. 30m_0615_refine.jsonl：含中英双语的 30m token 的数据集，英文占比为 0.615

这里，10m_en_refine.jsonl 和 10m_zh_refine.jsonl 是为了给最终模型提供训练数据进行预备的，30m_0615_refine.jsonl 是为了下一步进行指导模型的训练进行预备的。30m_0615_refine.jsonl 中的英文占比 0.615 为超参数，为通过实验选取的数值。

2.2 指导模型训练

基于原提交指南 **4. 训练**的 train_scripts/deepspeed_train_7b_lora.sh 对 30m_0615_refine.jsonl 进行训练，得到指导模型 30m_0615_baichuan。

2.3 数据筛选

2.3.1 原模型与指导模型 entropy 计算

为了对数据进行 entropy 计算，这里我们采用了 SelfCheckGPT[3] 中的计算方式：

$$H_j = -\sum_{w \in W} p_j(w) \log p_j(w),$$

$$Avg(H) = \frac{1}{J} \sum_j H_j.$$

这里，我们计算的是每一个数据样本对每一个 token 上的平均熵 $Avg(H)$，$p_j(w)$ 代表一个数据样本中第 j 个 token 为 w 的概率，W 为包含 w 的整个 token 词表，J 为数据样本中 token 的个数。

由此，我们可以分别计算 10m_en_refine.jsonl 和 10m_zh_refine.jsonl 中数据在原始 baichuan2-7b 模型上的 entropy 和 30m_0615_baichuan 模型上的 entropy，并用于后续数据筛选。

2.3.2 数据筛选

在前一步的基础上，分别对 10m_en_refine.jsonl 和 10m_zh_refine.jsonl 中数据进行筛选，保留指导模型上的熵 $Avg_{guide}(H)$ 小于原始模型上的熵 $Avg_{origin}(H)$ 的数据，得到 10m_en_entropy.jsonl 和 10m_zh_entropy.jsonl。

2.3.3 数据再采样

采用原提交指南 **3.2 数据集采样**的 get_train_dataset_7b.py 工具对 10m_en_entropy.jsonl 和 10m_zh_entropy.jsonl 再次进行采样得到 10m_05_entropy.jsonl。这里，我们严格按照比赛设置为 token 采样数量为 10M，中英文数据比例为 0.5。不过，我们也在实验中探究了不同数据比例的效果。

2.4 最终模型训练

基于原提交指南 **4. 训练**的 train_scripts/deepspeed_train_7b_lora.sh 对 10m_05_entropy.jsonl 进行训练，得到最终的模型 10m_05_entropy_baichuan。

3 实验

本部分对实验设置进行说明，并展示整体的方案效果，训练指导模型时不同数据比例混合的效果，以及最终模型训练时不同数据比例混合的效果。

3.1 实验设置

在训练模型部分，本实验完全基于比赛的设置进行。在训练指导模型时不同数据比例混合的展示时，为了进行大量实验寻找更好的中英文混合比例，其实际结果是在不同数据比例混合的 10M 的 token 上进行训练的，并推广在 30M 的 token 上训练指导模型。

在数据部分，实验结果在比赛提供的 dev 和第一阶段的 borad 数据上分别展示结果，其指标为比赛要求的指标。在 dev 上，将展示 dev 中的 5 个数据集分别的结果和平均的结果；在 board 上由于是提交测试，将只展示平均的结果。

3.2 整体的方案效果

Table 1 整体的方案效果

	dev						board
Data	ma	mc	mc-zh	qmsumm	summ	average	average
Metrics	mc2	acc	acc	rougeL	rougeL		
Baseline	0.3817	0.48	0.5712	0.0582	0.0551	0.3092	-
Our model	0.4491	0.47	0.5726	0.1771	0.1343	0.3606	**0.3800**

结果从表格 1中可以看到我们的方法与没训练前的 Baseline 模型相比，在 ma，qmsumm 和 summ 三个任务上均得到了极大的提高，并且 board 上取得了极好的成绩。不过，我们也发现 mc 和 mc-zh 数据的提升不多，甚至为负向提升，这可能与训练数据中与该任务相关的数据的比例有关，也可能与模型已经习得的能力上界有关。

3.3 指导模型不同数据比例混合效果

指导模型不同数据比例混合效果（10M 的 token 大小）如表格 2所示，我们可以发现当英文占比 0.615 时，其在 board 上的效果最好。因此，我们也基于此结果选择我们指导模型 30M 的 token 的数据集的中英文混合比。

另外，我们还发现，dev 和 board 数据可能本身会存在分布不一致的情况，当我们在 dev 数据上效果提升时，board 数据上的效果反而会下降。反之，也存在 board 数据上的效果提升时，dev 数据上效果也不足。

3.4 最终模型不同数据比例混合效果

我们还测试了在用我们的方案分别筛选了中英文数据后，进行不同比例混合训练得到的最终模型的效果，如表格 3所示。我们可以发现，这里与筛选前数据比例不一样，当中英文比例 1：1 时，其效果反而最好，这可能暗示了我们的方案筛选数据后，其分布已经有了较大的变化，中英文的比例的确需要重新配置。

Table 2 指导模型不同数据比例混合效果

				dev			board
Data	ma	mc	mc-zh	qmsumm	summ	average	average
Ratio	mc2	acc	acc	rougeL	rougeL		
0.385	0.4397	0.47	0.5680	0.1698	0.1327	0.3561	-
0.5	0.4412	0.46	0.5726	0.1614	0.1309	0.3532	0.3756
0.6	0.4429	0.47	0.5694	0.1619	0.1354	0.3559	0.3720
0.615	0.4455	0.47	0.5719	0.1653	0.1381	0.3582	**0.3779**
0.618	0.4424	0.48	0.5764	0.1665	0.1346	0.3600	0.3728
0.8	0.4419	0.47	0.5684	0.1653	0.1390	0.3569	0.3697
1	0.4419	0.48	0.5642	0.1642	0.1381	0.3577	-

不过，我们可以发现与表格 2相比，我们筛选后再训练的模型效果，在 dev 上即使采用不同的数据比例，都有一定的提升，这说明我们的方案是具有通用性的。另外，我们仍然发现 dev 和 board 数据可能本身会存在分布不一致的情况，即 dev 数据上效果提升不代表 board 数据上的效果提升。

Table 3 最终模型不同数据比例混合效果

				dev			board
Data	ma	mc	mc-zh	qmsumm	summ	average	average
Ratio	mc2	acc	acc	rougeL	rougeL		
0.45	0.4516	0.48	0.5698	0.1781	0.1338	0.3627	0.3761
0.5	0.4491	0.47	0.5726	0.1771	0.1343	0.3606	**0.3800**
0.55	0.4477	0.48	0.5691	0.1758	0.1347	0.3615	0.3764
0.6	0.4481	0.48	0.5670	0.1703	0.1363	0.3603	0.3720
0.615	0.4468	0.48	0.5684	0.1739	0.1362	0.3611	0.3742
0.7	0.4465	0.49	0.5694	0.1788	0.1356	0.3641	0.3694

4 结论

本团队构建了基于 LLM 的熵计算进行自动数据筛选的方案，对 FT-Data-Ranker 7B 赛道进行有关数据的筛选和训练，并在取得极好的效果。同时，我们也发现该比赛中，dev 和 borad 数据存在一定的分布差异，可能需要进一步更新方案来进一步提高领域外泛化能力。

参 考 文 献

[1] BAICHUAN. Baichuan 2: Open large-scale language models[A/OL]. 2023. https://arxiv.org/abs/2309.10305.

[2] ZHOU H, LIU T, MA Q, et al. Lobass: Gauging learnability in supervised fine-tuning data[A]. 2023.

[3] MANAKUL P, LIUSIE A, GALES M J. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models[A]. 2023.