

PR #2264 完整报告

THUDM/slime

fix: preserve consecutive tool responses in Qwen3 SFT tokenization

合并时间: 2026-08-12 13:26

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2264>

执行摘要

- 一句话: 修复 Qwen3 SFT 连续 tool 响应 tokenization 不一致
- 推荐动作: 值得精读, 特别是了解 loss-mask 生成器如何处理工具消息的上下文依赖。设计决策值得关注: 最小化改动而非全量重构, 通过分组连续 tool 消息修复 bug, 并保留原有架构。

功能与动机

issue #2263 指出 Qwen3 多轮 SFT 在并行工具调用场景下, loss-mask 生成器将每个 tool 消息独立渲染, 导致 token 序列与推理时 chat template 渲染不一致, 训练 / 推理 tokenization 不匹配。PR body 强调 token IDs 被直接用作 sample.tokens, 影响实际训练输入。

实现拆解

1. 将 `gen_multi_turn_loss_mask_qwen3()` 中的 for 循环改为 while 循环, 并引入分组逻辑: 当检测到 `role == "tool"` 时, 收集连续的 tool 消息为一个组 `message_group`。
2. 在调用 `apply_chat_template()` 时, 传入 `message_group` (以及 `prefix_message` 或作为前缀), 使得连续 tool 响应被渲染在同一 user turn 中, 避免多余的 `<lim_endl><lim_startl>` user 边界。
3. 其余消息类型 (非 tool) 保持按条渲染, 不影响 assistant 消息的 loss mask 生成。
4. 循环索引 `i` 在每次迭代后更新为 `group_end`, 确保每个分组只处理一次, 时间复杂度仍为 $O(n)$ 。
5. 在 `tests/utils/test_loss_mask_type_qwen35.py` 中新增回归测试 `test_qwen3_matches_full_template_for_consecutive_tool_responses`, 验证 token ID 和 loss mask 与完整模板渲染一致。

关键文件:

- `slime/utils/mask_utils.py` (模块 Mask 工具; 类别 source; 类型 core-logic; 符号 `gen_multi_turn_loss_mask_qwen3`): 核心修复文件, 修改 `gen_multi_turn_loss_mask_qwen3` 的分组逻辑
- `tests/utils/test_loss_mask_type_qwen35.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `test_qwen3_matches_full_template_for_consecutive_tool_responses`): 新增回归测试, 验证连续 tool 响应的 token ID 和 loss mask 与完整模板渲染一致

关键符号: gen_multi_turn_loss_mask_qwen3

关键源码片段

slime/utils/mask_utils.py

核心修复文件, 修改 gen_multi_turn_loss_mask_qwen3 的分组逻辑

```
def gen_multi_turn_loss_mask_qwen3(
    self, messages: list[dict], tools: list[dict] = None
) -> tuple[list[int], list[int]]:
    all_loss_masks = []
    all_token_ids = []

    prefix_message = {"role": "user", "content": "FOR CALCULATING LOSS MASK ONLY"}
    prefix_token_ids = self.tokenizer.apply_chat_template([prefix_message], tokenize=True,
return_dict=False)

    i = 0
    while i < len(messages):
        message = messages[i]
        if message["role"] == "tool":
            # Qwen 模板会将连续 tool 响应放在同一个 user turn 中。
            # 因此将连续 tool 消息收集为一组, 一次性传给 apply_chat_template,
            # 防止在响应之间插入多余的 user turn 边界。
            group_end = i + 1
            while group_end < len(messages) and messages[group_end]["role"] == "tool":
                group_end += 1
            message_group = messages[i:group_end]
        else:
            group_end = i + 1
            message_group = [message]

        if i == 0:
            # 第一条消息需要附加 prefix_message 以剥离最后生成的 token。
            tailed_message_ids = self.tokenizer.apply_chat_template(
                message_group + [prefix_message],
                tokenize=True,
                tools=tools,
                return_dict=False,
            )
            message_ids = tailed_message_ids[:-len(prefix_token_ids)]
        else:
            # 非首条消息在前面加上 prefix_message, 以便将其 token 剥离, 只保留本组消息的
            # token。
            prefixed_message_ids = self.tokenizer.apply_chat_template(
                [prefix_message] + message_group,
                tokenize=True,
                return_dict=False,
            )
```

```

        message_ids = prefixed_message_ids[len(prefix_token_ids):]

    if message["role"] != "system" and i > 0:
        message_ids = message_ids[self.system_message_length:]

    if message["role"] == "assistant":
        loss_mask = [0] * self.gen_token_length + [1] * (len(message_ids) - self.gen_token_length)
    else:
        loss_mask = [0] * len(message_ids)

    if message.get("step_loss_mask", 1) != 1:
        loss_mask = [0] * len(message_ids)

    all_loss_masks.extend(loss_mask)
    all_token_ids.extend(message_ids)
    i = group_end

return all_token_ids, all_loss_masks

```

评论区精华

无 review 评论或讨论。PR 描述中说明了设计权衡：不切换到完整对话渲染，而是仅对连续 tool 消息分组，以最小化行为变化。

- 暂无高价值评论线程

风险与影响

- 风险：影响核心训练数据生成路径，若分组逻辑有误（例如边界条件处理不当）可能导致 token 序列错乱，但测试覆盖了典型场景。改动仅涉及 `slime/utils/mask_utils.py` 中 Qwen3 的 loss-mask 生成，不影响 Qwen3.5 等其他路径。潜在风险：极端情况下工具消息之间夹杂其他角色时分组逻辑不适用（但符合预期），且 `apply_chat_template` 的渲染行为依赖 tokenizer 实现，未来 transformers 升级可能改变行为。
- 影响：影响所有使用 Qwen3 SFT 且包含并行工具调用（连续 tool 消息）的训练任务，修复后训练与推理 tokenization 一致，提升模型行为一致性和训练质量。对单条 tool 响应或非连续 tool 响应无影响。团队需关注是否有依赖旧行为的后处理逻辑，但预期无。
- 风险标记：核心路径变更，测试覆盖有限

关联脉络

- 暂无明显关联 PR