

PR #1960 完整报告

THUDM/slime

Extract more util code from coding_agent_rl example

合并时间: 2026-05-27 15:13

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1960>

执行摘要

- 一句话: 抽取 coding_agent_rl 通用工具到核心 agent 模块
- 推荐动作: 建议代码审查者重点关注 trajectory.py 中 merge_turns 的多轮对齐逻辑和 parsing.py 的工具调用解析路径, 确保与原有行为一致。值得阅读的设计决策包括将配置从环境变量迁移到 args 的统一管理方式。

功能与动机

延续近期将示例代码迁移到 slime/agent 核心库的系列工作 (PR #1956、#1958、#1957), 进一步抽离通用工具函数和数据结构, 避免重复实现, 降低维护成本。

实现拆解

1. 创建 `slime/agent/trajectory.py`: 定义 TurnRecord 和 TokenSegment 不可变数据类, 实现 merge_turns 将多轮对话线性化为单段训练样本、write_segment_to_sample 填充 Sample 字段、fan_out_sample_segments 将多段样本展平并均分奖励。
2. 创建 `slime/agent/parsing.py`: 定义 ParsedModelOutput 数据类, 提供 parse_model_output 统一调用 SGLang 的 reasoning 和 function-call 解析器, 并保留 XML 回退解析。
3. 修改 `examples/coding_agent_rl/middleware.py`: 移除内联的 _update_prompt_and_mask 和 verify_tito_cross_turn, 改为使用 slime.agent.trajectory 中的 TurnRecord 和 merge_turns; 简化 _build_prompt 和 _generate 接口, 将 prompt/output 记录为 TurnRecord 列表, 后续由 pop_session_split 线性化。
4. 修改 `examples/coding_agent_rl/generate.py`: 删除内联的 _write_segment_to_sample 和 _fan_out_to_samples, 改从 slime.agent.trajectory 导入 fan_out_sample_segments; 移除 SWE_MAX_RESPONSE_TOKENS、SWE_TOOL_PARSER 等环境变量, 改为从 args 获取 rollout_max_context_len、sglang_tool_call_parser 等。
5. 更新启动脚本和文档: 删除 SWE_MAX_RESPONSE_TOKENS、SWE_MAX_SEGMENT_TOKENS、SWE_SAVE_TRAJECTORY_TREE 等环境变量导出, 在 README 中说明新增的命令行参数用法。
6. 无对应测试文件改动, 风险在于新模块缺少测试覆盖。

关键文件:

- `slime/agent/trajectory.py` (模块 Agent 核心; 类别 source; 类型 core-logic; 符号 `TurnRecord`, `TokenSegment`, `_prompt_matches_current`, `_output_mask`) : 核心新增文件, 定义 agent 轨迹转训练样本的数据结构与算法, 包括 `TurnRecord`、`TokenSegment`、`merge_turns`、`fan_out_sample_segments` 等关键符号。
- `slime/agent/parsing.py` (模块 Agent 核心; 类别 source; 类型 dependency-wiring; 符号 `ParsedModelOutput`, `parse_model_output`, `parse_tool_uses`, `parse_xml_tool_uses`) : 新增模型输出解析模块, 封装 SGLang 的 reasoning/function-call 解析及 XML 回退, 供所有 agent workflow 复用。
- `examples/coding_agent_rl/middleware.py` (模块 Agent 编排; 类别 source; 类型 dependency-wiring; 符号 `_render_token_ids`, `_common_prefix_len`, `verify_tito_cross_turn`, `_update_prompt_and_mask`) : 主要修改的示例文件, 移除内联轨迹处理逻辑, 改为调用 `slime.agent.trajectory`, 是重构的关键环节。
- `examples/coding_agent_rl/generate.py` (模块 Agent 编排; 类别 source; 类型 dependency-wiring; 符号 `_write_segment_to_sample`, `_fan_out_to_samples`, `_pop_segments`) : 移除内联样本组装和环境变量, 改为从核心模块导入 `fan_out_sample_segments` 并通过 `args` 传递配置。
- `examples/coding_agent_rl/run_qwen36_35b_a3b_swe_8nodes.sh` (模块 脚本配置; 类别 other; 类型 core-logic) : 移除不再使用的环境变量导出, 反映配置迁移。
- `examples/coding_agent_rl/README.md` (模块 文档; 类别 docs; 类型 documentation) : 文档更新, 反映配置和接口变更。

关键符号: `TurnRecord`, `TokenSegment`, `merge_turns`, `write_segment_to_sample`, `fan_out_sample_segments`, `ParsedModelOutput`, `parse_model_output`, `parse_tool_uses`, `parse_xml_tool_uses`, `_render_token_ids`, `_common_prefix_len`, `verify_tito_cross_turn`, `_update_prompt_and_mask`, `_generate`, `_record_turn`

关键源码片段

`slime/agent/trajectory.py`

核心新增文件, 定义 agent 轨迹转训练样本的数据结构与算法, 包括 `TurnRecord`、`TokenSegment`、`merge_turns`、`fan_out_sample_segments` 等关键符号。

```
"""Token-level trajectory helpers for agent rollouts."""
```

```
from __future__ import annotations
import copy
import dataclasses
import logging
from typing import Any
from slime.utils.types import Sample
```

```
logger = logging.getLogger(__name__)
```

```
@dataclasses.dataclass(frozen=True)
```

```
class TurnRecord:
```

```
    """一次模型生成回合的精确 token 快照。
```

``prompt\_ids`` 是发送给生成器的完整 token 化 prompt。
``output\_ids`` 是原始生成的输出。
``output\_loss\_mask`` 默认全 1，可被每回合验证清零。
"""

```
prompt_ids: list[int]
output_ids: list[int]
output_loss_mask: list[int]
finish_reason: str
```

```
@dataclasses.dataclass(frozen=True)
```

```
class TokenSegment:
```

```
    """从 agent 轨迹组装的一个训练片段。"""
```

```
    prompt_ids: list[int]
```

```
    response_ids: list[int]
```

```
    loss_mask: list[int]
```

```
    metadata: dict[str, Any] = dataclasses.field(default_factory=dict)
```

```
def merge_turns(turns: list[TurnRecord], *, metadata: dict[str, Any] | None = None) ->
TokenSegment | None:
```

```
    """将 TurnRecord 列表重放为单个线性 TokenSegment。
```

第一轮的 prompt 作为片段 prompt。后续轮的 prompt 必须以
``prompt + response\_so\_far`` 开头；其后缀是新非模型上下文（loss_mask=0），
接着是模型输出及其每回合输出 mask。

```
    """
```

```
if not turns:
```

```
    return None
```

```
prompt_ids = list(turns[0].prompt_ids)
```

```
response_ids: list[int] = []
```

```
loss_mask: list[int] = []
```

```
for i, turn in enumerate(turns):
```

```
    if i > 0:
```

```
        expected_len = len(prompt_ids) + len(response_ids)
```

```
        if turn.prompt_ids[:len(prompt_ids)] == prompt_ids and \
```

```
            turn.prompt_ids[len(prompt_ids):expected_len] == response_ids:
```

```
            # 正常追加: prompt 是已有 prompt+response, 多出的部分是新的非模型上下文
```

```
            context_tail = turn.prompt_ids[expected_len:]
```

```
            response_ids.extend(context_tail)
```

```
            loss_mask.extend([0] * len(context_tail))
```

```
        elif turn.prompt_ids[:len(prompt_ids)] == prompt_ids:
```

```
            # prompt 前缀匹配但后续偏移, 重新基线化
```

```
            logger.warning("[trajectory] merge prefix drift; rebaselining segment")
```

```
            response_ids = list(turn.prompt_ids[len(prompt_ids):])
```

```
            loss_mask = [0] * len(response_ids)
```

```
        else:
```

```
            # prompt 基础改变, 从当前 prompt 重新开始
```

```
            logger.warning("[trajectory] merge prompt base changed; starting fresh")
```

```
            prompt_ids = list(turn.prompt_ids)
```

```
            response_ids = []
```

```
        loss_mask = []
        response_ids.extend(turn.output_ids)
        loss_mask.extend(turn.output_loss_mask if len(turn.output_loss_mask) == len(turn.output_ids)
                           else [0]*len(turn.output_ids))
    return TokenSegment(prompt_ids=prompt_ids, response_ids=response_ids,
                        loss_mask=loss_mask, metadata=dict(metadata or {}))
```

评论区精华

无 review 评论或讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 回归风险：middleware.py 和 generate.py 重构幅度大（+161/-260、+34/-103），轨迹记录和样本组装逻辑改为调用新模块，可能出现边界情况处理不一致。
 2. 缺少测试覆盖：trajectory.py 和 parsing.py 均为新增且无单元测试，merge_turns 的多轮线性化逻辑容易出错。
 3. 配置兼容性：generate.py 从环境变量改为 args 获取 parser 参数，若调用方未通过 args 传递正确值，可能导致解析失败。
- 影响：
 1. 用户：示例代码配置变简洁，需改用 --rollout-max-context-len 等统一参数。
 2. 系统：核心 agent 模块增加 trajectory 和 parsing 基础设施，为更多 agent workflow 提供复用基础。
 3. 团队：维护点从示例内联代码转移到核心模块，降低重复。 - 风险标记：核心路径变更，缺少测试覆盖，配置接口变更

关联脉络

- PR #1956 Add slime/agent/ and move sandbox impl inside: 同一系列工作，将 sandbox 从示例迁移至核心模块。
- PR #1958 Move coding_agent_rl's helper function to sandbox.py: 同一系列工作，移动辅助函数至 sandbox。
- PR #1957 Minor refactor for coding agent rl logic and remove SWE_LIST_TRAJECTORY: 上一个重构 PR，配置变更与本 PR 有连续性。
- PR #1923 [examples] add coding_agent_rl: agent-in-sandbox RL minimal demo: 最初引入 coding_agent_rl 示例的 PR，本 PR 从中提取代码。