PR #25016 完整报告

sgl-project/sglang

[bench] Agentic support for `bench_serving.py`

合并时间：2026-05-13 07:00

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25016

执行摘要

- 一句话：bench_serving 新增 agentic 多消息轮次支持
- 推荐动作：建议合并。变更小巧清晰，扩展了基准测试能力，且向下兼容。后续可补充针对新格式的单元测试。

功能与动机

为了回放 agentic / 工具使用工作负载（例如 SWE-Smith），需要每轮支持多条消息（用户消息 + 多条 tool 观察），而不是仅一条用户字符串。

实现拆解

1. 在 bench_serving.py 中新增 _normalize_round_messages(turn) 函数，将单个字符串或 List[Dict] 消息列表统一为消息字典列表；异常形状返回 None。
2. 重构 wrap_multi_turn_request_func 中的循环：使用 _normalize_round_messages 处理每轮输入，若返回 None 则抛出明确的 ValueError，并使用 prev_messages.extend(normalized) 替代原有的单条追加。
3. 更新 benchmark() 中的 multi-turn 检测逻辑：使用 _normalize_round_messages(first_prompt[0]) is not None 替代简单的字符串类型检查，从而正确区分单次 OpenAI 消息列表（List[Dict]）和 multi-turn 输入。
4. 在 autobench.py 的 _normalize_prompt 中新增一条分支：当 prompt 是 List[List[Dict]]（每个元素为包含 role/content 的 dict 的列表）时，返回 multi_turn 类型，以便加载每轮多条消息的数据集。

关键文件：

- python/sglang/bench_serving.py（模块 基准测试；类别 source；类型 core-logic；符号 _normalize_round_messages, wrap_multi_turn_request_func, benchmark）：核心变更文件，新增 _normalize_round_messages 函数并重构 multi-turn 逻辑
- python/sglang/benchmark/datasets/autobench.py（模块 数据集处理；类别 source；类型 core-logic；符号 _normalize_prompt）：新增对每轮多条消息的 prompt 格式的识别，用于加载 agentic 数据集

关键符号：_normalize_round_messages, wrap_multi_turn_request_func, _normalize_prompt

关键源码片段

python/sglang/bench_serving.py

核心变更文件，新增 _normalize_round_messages 函数并重构 multi-turn 逻辑

```
def _normalize_round_messages(turn: Any) -> Optional[List[Dict[str, str]]]:
    """Normalize a multi-turn round to a list of message dicts.

    Accepts ``str`` (single user message) or ``List[Dict]`` with role/content
    (e.g. multiple tool observations bundled into one round). Returns ``None``
    on any other shape so callers can also use it as a predicate.
    """
    if isinstance(turn, str):
        return [{"role": "user", "content": turn}]
    if (
        isinstance(turn, list)
        and turn
        and all(isinstance(m, dict) and "role" in m and "content" in m for m in turn)
    ):
        return [{"role": m["role"], "content": m["content"]} for m in turn]
    return None

# 在 wrap_multi_turn_request_func 中使用 :
for round_index in range(len(prompts)):
    normalized = _normalize_round_messages(prompts[round_index])
    if normalized is None:
        raise ValueError(
            f"Multi-turn round {round_index} must be a str or a "
            "non-empty List[Dict] of role/content messages, got: "
            f"{type(prompts[round_index]).__name__}"
        )
    prev_messages.extend(normalized)
    # ... 后续处理
```

评论区精华

无实质性讨论；只有一条机器人自动评论和一条 CI 触发命令，Qiaolin-Yu 直接批准。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更集中在基准测试工具，不影响核心推理流程。主要风险是新的 multi-turn 格式可能与其他已有数据集格式（如单次 OpenAI 消息、token_ids 等）混淆，但检测逻辑已通过 _normalize_round_messages 严格区分。未添加单元测试是潜在风险。
- 影响：影响范围：仅 bench_serving CLI 工具和 autobench 数据集加载。用户可以使用新的 List[List[Dict]] 格式定义每轮包含多条消息的数据集，从而模拟更真实的 agentic 场景。与现有 List[str] 格式和单次 List[Dict] 格式完全兼容。

- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR