

PR #44551 完整报告

vllm-project/vllm

fix: Correct reasoning-end detection for prompt history

合并时间: 2026-06-27 13:15

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/44551>

执行摘要

- 一句话: 修复 Cohere 推理结束检测, 避免历史 prompt 误判
- 推荐动作: 值得精读, 特别是 `is_reasoning_end` 的重构逻辑, 展示了如何通过状态扫描将范围限定在当前 turn。设计决策值得参考。

功能与动机

PR body 指出: 原始实现简单扫描整个输入是否有 `<IEND_THINKINGI>`, 并且可以在初始 turn 或中间 delta 上调用。这意味着它会把旧的 thinking token (包括系统 turn 中提到的) 当作 thinking 结束的信号, 导致解析失败并返回特殊 token。

实现拆解

1. 在 `BaseCohereCommandReasoningParser.__init__` 中新增 `self.chatbot_token_id` 和 `self.start_token_id` (原来只有 `self.end_token_id`) 。
2. 重写 `is_reasoning_end` 方法: 从后向前遍历 `input_ids`, 遇到 `<ISTART_THINKINGI>` 则返回是否看到过 end token; 遇到 `<ICHATBOT_TOKENI>` 则返回 `False` (表示未结束); 遇到 `<IEND_THINKINGI>` 则设置标志。
3. 添加单元测试文件 `tests/reasoning/test_cohere_command_reasoning_parser.py`, 包含 `MockCohereTokenizer` 和多个 `ReasoningCase` 测试用例, 覆盖 `cmd3` 和 `cmd4` 解析器、单工具调用和带 emoji 的引用情况。
4. 在 `requirements/test/cuda.in` 和 `requirements/test/cuda.txt` 中添加 `cohere_melody` 依赖, 以确保测试可用。

关键文件:

- `vllm/reasoning/cohere_command_reasoning_parser.py` (模块 推理解析; 类别 source; 类型 core-logic; 符号 `BaseCohereCommandReasoningParser.init`, `BaseCohereCommandReasoningParser.is_reasoning_end`): 核心逻辑变更: 修改 `__init__` 存储额外 token id, 重写 `is_reasoning_end` 方法实现基于 turn 边界的检测。
- `tests/reasoning/test_cohere_command_reasoning_parser.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `ExpectedToolCall`, `ReasoningCase`, `MockCohereTokenizer`, `REASONING_CASES`): 新增 625 行测试, 覆盖 `cmd3` 和 `cmd4` 解析器的单工具调用、带 emoji 的引用等场景, 并包含 `MockCohereTokenizer` 模拟 tokenizer。

- requirements/test/cuda.in (模块 测试依赖; 类别 test; 类型 test-coverage) : 添加 `cohere_melody>=0.9.0` 依赖, 用于 Cohere 推理解析器测试中的 melody 过滤库。
- requirements/test/cuda.txt (模块 依赖锁定; 类别 docs; 类型 documentation) : 锁定 `cohere-melody==0.9.0` 版本, 作为 CUDA 测试环境的依赖。

关键符号: `BaseCohereCommandReasoningParser.init`,
`BaseCohereCommandReasoningParser.is_reasoning_end`,
`MockCohereTokenizer.convert_tokens_to_ids`, `MockCohereTokenizer.get_vocab`,
`MockCohereTokenizer.encode`, `MockCohereTokenizer.decode`

评论区精华

sfeng33 reviewer 要求添加单元测试 (流式和非流式), 作者询问可不可以添加 melody 作为测试依赖, 最终测试被添加。另外有 pre-commit 失败和测试失败的评论 (`KeyError: 'cohere2_moe'`), 但最终被修复并由 sfeng33 批准合并。

- 添加单元测试覆盖流式和非流式场景 (testing): 作者添加了测试文件, 并询问是否可添加 melody 依赖, 最终被接受。
- pre-commit 和 CI 测试失败 (other): 作者修复了问题, 最终 CI 通过。

风险与影响

- 风险: 风险较低, 因为变更集中在特定的 Cohere 推理解析器, 且添加了全面测试。可能的回归: 如果其他代码依赖旧的 `is_reasoning_end` 行为 (全局扫描), 但该方法内部使用的, 影响范围小。性能影响: 新方法遍历整个 `input_ids`, 与原来类似, 但增加了条件判断, 影响微乎其微。
- 影响: 用户: 修复了 Cohere Command 模型在对话历史包含旧 thinking token 时的解析错误, 提高可靠性。系统: 无性能影响, 无架构变更。团队: 增加了测试覆盖, 有利于未来维护。
- 风险标记: 核心路径变更, 测试覆盖已补充

关联脉络

- 暂无明显关联 PR