

PR #48034 完整报告

vllm-project/vllm

[Rust Frontend] Tolerate whitespace before the outer brace in JSON tool-call parsers

合并时间: 2026-07-15 19:03

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48034>

Rust JSON 工具调用解析器空白容差修复

执行摘要

该 PR 修复了 Rust 前端 JSON 工具调用解析器的一个行为不一致问题: 当工具调用 JSON 参数对象闭花括号与外部闭花括号之间存在空白时, 解析器会失败。通过在外层 `}` 前添加 `ws0` 跳过可选空白, 使其与 Python `json.loads` 行为对齐。修改涉及两个解析核心文件, 并补充了单元测试和集成测试。

功能与动机

当模型输出 pretty-printed 格式的 JSON (例如 `{"name": "f", "parameters": {"x": 1}}`) 时, 外层 `}` 前存在空格。Python 标准库 `json.loads` 能够容忍这种空白, 但 Rust 自定义解析器未做处理, 导致解析失败。Issue 中期望修复以增强鲁棒性并减少因模型输出格式波动导致的错误。

实现拆解

- 通用 JSON 解析器 (`rust/src/parser/src/tool/json/mod.rs`): 在 `tool_call_close_event` 中将 `literal("{}")` 替换为 `seq!(_: ws0, _: literal("{}"))`, 使解析器在匹配闭合花括号前先跳过空白。
- Llama JSON 解析器 (`rust/src/parser/src/tool/json/llama.rs`): 对 `tool_call_close_event` 执行相同修改, 保持两种解析器行为一致。
- 单元测试: 为两个解析器分别新增测试函数, 验证完整输入和跨分块输入均能正确解析。特别测试了空白在分块边界的情况, 确保 `Partial` 模式下 `ws0` 的 `Incomplete` 处理正确。
- 集成测试 (`rust/src/chat/src/output/default/unified.rs`): 添加 `unified_stream_pares_formatted_tool_call_without_latch`, 模拟流式场景: 先输入带空格的工具调用, 再输入正常工具调用, 确认首个格式化调用不会触发 `parse-error latch` 导致后续调用被错误转为文本。

`rust/src/parser/src/tool/json/mod.rs`

通用 JSON 工具调用解析器的核心修复文件, 修改 `tool_call_close_event` 以跳过外部花括号前空白, 并添加对应的单元测试。

```
/// 解析 marker 包裹的 JSON 工具调用的闭合标记。
/// 现在在解析外部 `}` 前跳过可选空白, 以与 Python json.loads 行为对齐。
fn tool_call_close_event(
    input: &mut JsonToolInput<'_,>,
```

```

    config: JsonToolCallConfig,
) -> ModalResult<JsonToolCallEvent> {
    // seq! 组合子依次执行 ws0 ( 跳过空白 ) 和 literal("{}") ( 匹配闭合花括号 )
    seq!(_: ws0, _: literal("{}")).parse_next(input)?;

    match config.delimiter {
        Some(delimiter) => alt((
            input: &mut JsonToolInput<'_>| tool_call_end_event(input, config),
            input: &mut JsonToolInput<'_>| tool_call_delimiter_event(input, delimiter),
        ))
        .parse_next(input),
        None => tool_call_end_event(input, config),
    }
}

```

rust/src/parser/src/tool/json/llama.rs

Llama JSON 工具调用解析器的核心修复文件，同样修改 `tool_call_close_event` 添加空白跳过，并添加回归测试。

```

/// 解析一个 Llama JSON 工具调用的外部闭合花括号。
/// 前加 ws0 以跳过可能存在的空白字符，保持与通用解析器行为一致。
fn tool_call_close_event(input: &mut JsonToolInput<'_>) -> ModalResult<LlamaJsonEvent> {
    seq!(_: ws0, _: literal("{}"))
        .value(LlamaJsonEvent::ToolCallClose)
        .parse_next(input)
}

```

评论区精华

无实质性技术讨论。BugenZhao 审核并批准。

风险与影响

- 风险：低。修改范围极小，仅在两个解析函数的闭合花括号匹配前增加空白跳过。测试覆盖了正常和分块场景，避免了回归。
- 影响：增强 Rust 前端的工具调用解析健壮性，减少因模型输出格式波动导致的失败。用户无需更改任何配置即可受益。

关联脉络

该 PR 是与 Python 解析器行为对齐的持续改进的一部分。此前已有其他修复确保流式解析器正确处理分块边界（如 `ws0` 与 `Partial` 模式的兼容）。无直接关联的历史 PR，但该修复提升了整体工具调用可靠性。