

PR #45025 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Stop unescaping XML-style tool-call parameter values

合并时间: 2026-06-11 10:05

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45025>

执行摘要

本次 PR 修复了 Rust 前端中四个工具调用解析器（minimax_m2、qwen_coder、glm_xml、deepseek_dsml）对参数值执行 XML unescape 导致数据损坏的 bug。删除整个 `xml_unescape` 函数及相关调用后，参数值保持模型输出的原始文本，与 Python 前端行为对齐。该修复安全、无副作用。

功能与动机

PR 描述明确指出渲染器（如 DeepSeek DSML renderer、MiniMax-M2 chat template）将字符串参数值原样输出，不进行 XML 转义；但解析器却误用 `xml_unescape` 对这些值进行解码，导致 `&` 变成 `&`、`<` 变成 `<` 等数据损坏。例如 `write_file` 参数 `Tom & Jerry <3` 被错误解码为 `Tom&Jerry<3`。Python 前端不做 unescape，因此两个前端行为不一致。修复后解析器保留原始实体，确保一致性。

实现拆解

1. 删除核心函数：在 `rust/src/tool-parser/src/utils.rs` 中移除 `xml_unescape` 和 `decode_xml_entity` 函数及其五个单元测试，同时删除不再需要的 `use std::borrow::Cow` 导入。
2. 修改四个解析器：在 `minimax_m2.rs`、`glm_xml/mod.rs`、`qwen_coder.rs`、`deepseek_dsml/mod.rs` 中移除对 `xml_unescape` 的导入，并删除参数值解析链中的 `.map(xml_unescape)` 调用，直接使用 `take_until` 返回的原始字符串转换为 `String`。
`deepseek_dsml/deepseek_v32.rs` 中改用 `str::to_string`。
3. 更新测试逻辑：在对应解析器的测试模块中，将原本断言 unescape 后内容的测试改为断言原始实体（如 `&`、`<`）被保留；新增 `preserves_raw_entities` 测试验证文字实体不转义。测试名称也同步修改，反映“保留原始内容”的意图。

关键源码片段

`rust/src/tool-parser/src/utils.rs`

核心文件，删除 `xml_unescape` 和 `decode_xml_entity` 函数及其测试，是整个修复的基础。

```
// rust/src/tool-parser/src/utils.rs
// 删除 xml_unescape 和 decode_xml_entity 后，不再需要 std::borrow::Cow
use winnow::error::{ContextError, ErrMode, ModalResult, Needed, StrContext, StrContextValue};
```

```

use winnow::stream::{Offset, Partial, Stream};

use super::Result;

// (partial_prefix_len、safe_text_len 等函数保持不变 )

/// Streaming lexical state for a top-level JSON object.
#[derive(Debug, Clone, Default, PartialEq, Eq)]
pub(super) struct JsonObjectScanState {
    // ...
}

// [ 删除约 120 行, 包括 xml_unescape 和 decode_xml_entity 函数以及五个测试 ]

```

rust/src/tool-parser/src/minimax_m2.rs

MiniMax-M2 解析器, 参数值解析中移除了 xml_unescape, 并更新测试验证原始实体保留。

```

// rust/src/tool-parser/src/minimax_m2.rs

/// 解析 MiniMax M2 参数块: 不再对参数值进行 xml_unescape, 保持原始内容
fn parameter(input: &mut &str -> ModalResult<(String, String)> {
    let (name, value) = seq!(
        _: literal(PARAMETER_START),
        _: (ws1, literal("name=")),
        attr_value,
        _: literal(">"),
        take_until(0.., PARAMETER_END), // 直接取原始字符串, 不再 .map(xml_unescape)
        _: literal(PARAMETER_END),
    )
    .parse_next(input)?;
    Ok((name.trim().to_string(), value.to_string()))
}

// 测试: 确认原始实体 (如 &amp;) 被保留, 不转义
#[test]
fn minimax_m2_parse_complete_preserves_raw_entities_in_parameter_value() {
    let mut parser = MinimaxM2ToolParser::new(&test_tools());
    let output = parser
        .parse_complete(&build_tool_block(&[(
            "get_weather",
            vec![(("city", "Tom & Jerry <3")],
        ))))
        .unwrap();
    let args: Value = serde_json::from_str(&output.calls[0].arguments).unwrap();
    assert_eq!(args["city"], json!("Tom & Jerry <3"));
}

```

rust/src/tool-parser/src/glm_xml/mod.rs

GLM 解析器, 参数值解析中移除 xml_unescape, 更新对应测试。

```
// rust/src/tool-parser/src/glm_xml/mod.rs

/// 解析 GLM 参数键值对：不再对参数值进行 xml_unescape
fn parse_parameter(input: &mut &str) -> ModalResult<(String, String)> {
    let (key, value) = seq!(
        _: literal(ARG_KEY_START),
        take_until(1.., ARG_KEY_END),
        _: literal(ARG_KEY_END),
        _: ws0,
        _: literal(ARG_VALUE_START),
        take_until(0.., ARG_VALUE_END).map(str::trim), // 仅修剪空白，不转义
        _: literal(ARG_VALUE_END),
    )
    .parse_next(input)?;
    Ok((key.trim().to_string(), value.to_string()))
}
```

评论区精华

审阅者 BugenZhao 简洁地总结："I agree that as long as there's no escaping in the chat template, we should not assume that the model will escape the arguments spontaneously. Removing unescaping in tool parser LGTM." 该评论确认了设计前提（chat template 不做转义）并认可移除 unescape 的修复方向。

风险与影响

风险：极低。仅移除错误的 unescape 操作，不引入新逻辑或其他变更路径。所有修改均在工具解析器内部，不影响其他模块。

影响：影响范围限定在使用 Rust 前端且使用这四种解析器的场景。用户参数值将不再被错误解码，返回模型输出的原始文本；与 Python 前端行为完全一致。用户无需任何配置迁移。

关联脉络

此 PR 与 Python 端 #44060（截断原始关闭标签）共享同一问题根源——chat template 输出原始内容，解析器需要正确处理。两者共同完善了 vLLM 前端对工具调用参数原始文本的处理策略，确保跨前端一致性。