PR #43243 完整报告

vllm-project/vllm

fix: parse Qwen3 XML JSON arguments first

合并时间：2026-05-28 11:35

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43243

执行摘要

- 一句话：修复 Qwen3 XML 参数解析中 JSON 布尔 /null 失败
- 推荐动作：值得精读该 PR 的处理方式：它展示了一种在不破坏向后兼容的前提下修复非标准输入解析问题的实用技巧——优先使用更严格 / 标准的解析器，再 fallback 到宽松的解析器。对于其他 tool parser 的类似问题（如 DeepSeek 或 Mistral 解析器）可参考此模式。

功能与动机

Issue #43238 报告 qwen3xml_tool_parser 在解析包含 JSON 布尔值（false）和 null 的数组参数时，ast.literal_eval 转换失败并退化为字符串输出，导致下游工具调用无法得到正确的原生数组。PR 的解决方案是在 deferred 参数分支中优先使用 json.loads，仅在 JSON 解析失败时回退到 ast.literal_eval。

实现拆解

1. 修改核心解析逻辑（vllm/tool_parsers/qwen3xml_tool_parser.py）：在 StreamingXMLToolCallParser._end_element 方法的 deferred 参数解析分支中，将原单次 ast.literal_eval(raw_for_parse) 替换为双重尝试：先执行 json.loads(raw_for_parse)，若抛出 json.JSONDecodeError 则再执行 ast.literal_eval(raw_for_parse)。这一调整确保 JSON 格式的字面量（如 false、null）能被准确解析，同时保留了对 Python 字面量格式的向后兼容。外层 except Exception 仍然作为最终 fallback 输出原始字符串。
2. 新增测试夹具与回归测试（tests/tool_parsers/test_qwen3coder_tool_parser.py）：
   - 新增 QUESTION_PARAMS 夹具，定义包含布尔字段 multiSelect 和可能为 null 的 answer 字段的数组参数结构。
   - 新增导入 StreamingXMLToolCallParser。
   - 新增测试函数 test_qwen3xml_deferred_array_parses_json_literals，构造一个包含 false 和 null 的数组参数 XML 输入，直接调用 parser.parse_single_streaming_chunks 并断言解析后的 DeltaMessage.tool_calls[0].function.arguments 正确反序列化为原生 Python 对象（False 和 None）。
3. 测试修复：首次 CI 失败后，作者将测试断言从内部 buffer 改为 parse_single_streaming_chunks 返回的 DeltaMessage 对象，避免因内部状态误解导致错误。

关键文件：

- vllm/tool_parsers/qwen3xml_tool_parser.py（模块 工具解析器；类别 source；类型 core-logic；符号 _end_element）：核心修复：在 deferred 参数解析中优先使用 json.loads，仅在其失败时回退至 ast.literal_eval。
- tests/tool_parsers/test_qwen3coder_tool_parser.py（模块 测试；类别 test；类型 test-coverage；符号 test_qwen3xml_deferred_array_parses_json_literals）：新增回归测试，覆盖数组参数包含 JSON false/null 的场景，并验证公开 API 返回值的正确性。

关键符号：_end_element

关键源码片段

vllm/tool_parsers/qwen3xml_tool_parser.py

核心修复：在 deferred 参数解析中优先使用 json.loads，仅在其失败时回退至 ast.literal_eval。

```
# vllm/tool_parsers/qwen3xml_tool_parser.py 第 817-832 行
try:
    # 若之前延迟了尾部换行，先补回
    if self.should_emit_end_newline:
        raw_for_parse = raw_text + "\n"
    else:
        raw_for_parse = raw_text
    try:
        # 优先使用 JSON 解析，以正确处理 false、null 等字面量
        parsed_value = json.loads(raw_for_parse)
    except json.JSONDecodeError:
        # 若 JSON 失败，回退至 Python 字面量解析（兼容旧模型输出）
        parsed_value = ast.literal_eval(raw_for_parse)
    output_arguments = json.dumps(parsed_value, ensure_ascii=False)
except Exception:
    # 最终 fallback：将原始文本作为字符串输出
    output_arguments = json.dumps(raw_text, ensure_ascii=False)
    parsed_value = raw_text
```

tests/tool_parsers/test_qwen3coder_tool_parser.py

新增回归测试，覆盖数组参数包含 JSON false/null 的场景，并验证公开 API 返回值的正确性。

```
# tests/tool_parsers/test_qwen3coder_tool_parser.py 新增部分
QUESTION_PARAMS = {
    "type": "object",
    "properties": {
        "questions": {
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "question": {"type": "string"},
                    "multiSelect": {"type": "boolean"},
                    "answer": {"type": "string"}, # 可为 null
```

```
            },
        },
    },
},
}

def test_qwen3xml_deferred_array_parses_json_literals():
    parser = StreamingXMLToolCallParser()
    parser.set_tools(
        [
            ChatCompletionToolsParam(
                type="function",
                function={"name": "AskUserQuestion", "parameters": QUESTION_PARAMS},
            )
        ]
    )

    # 模拟包含 JSON false 和 null 的模型输出
    delta = parser.parse_single_streaming_chunks(
        """<tool_call>
<function=AskUserQuestion>
<parameter=questions>
[{"question": "Pick a color", "multiSelect": false, "answer": null}]
</parameter>
</function>
</tool_call>"""
    )

    # 提取 arguments 字符串并验证 JSON 反序列化结果
    arguments = "".join(
        tool_call.function.arguments or ""
        for tool_call in delta.tool_calls or []
        if tool_call.function and tool_call.function.arguments is not None
    )
    assert json.loads(arguments) == {
        "questions": [
            {"question": "Pick a color", "multiSelect": False, "answer": None}
        ]
    }
```

评论区精华

gemini-code-assist[bot] 指出当前代码新增的 .strip() 是冗余的，因为 json.loads 原生处理前后空白。作者接受了该建议，并在后续提交中移除了 .strip() 调用。此外，sfeng33 在 review 中简单批准了该修复。

- 冗余 .strip() 调用 (style): 作者采纳建议，在后续提交中移除了 .strip()。

风险与影响

- 风险：风险极低：变更仅影响 deferred 参数解析分支，且为双重尝试（先 JSON 再 Python 字面量），外层 Exception 兜底保留。若模型输出来本是合法的 Python 字面量且被意外当作 JSON 解析成功，理论上可能改变语义（如 true 被解析为 True 而非字符串 "true"），但考虑到工具调用场景中参数值本就是结构化 JSON，此类冲突几乎不会发生。新增的测试覆盖了关键路径，进一步降低回归风险。
- 影响：直接修复了 Qwen3 XML 工具调用解析器在处理包含 JSON 布尔值或 null 的参数时的功能性 bug，影响所有使用 --tool-call-parser qwen3-xml 标志且工具参数包含 boolean 或 null 类型字段的用户。变更范围小（仅一个源文件 + 一个测试文件），对系统性能无显著影响。
- 风险标记：最小变更

关联脉络

- PR #43238 Bug: qwen3xml_tool_parser: ast.literal_eval fails on JSON booleans/null: 该 PR 正是为此 issue 提交的修复。