

PR #45795 完整报告

vllm-project/vllm

[Bugfix] Gemma4: skip forced JSON for required/named tool choice

合并时间: 2026-06-17 04:38

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/45795>

执行摘要

- 一句话: 修复 Gemma4 required/named tool choice 误返回 JSON
- 推荐动作: 建议阅读: 展示了通过子类重写 `adjust_request` 优雅绕过基类 JSON 约束的通用模式, 对其他模型解析器维护有参考价值。

功能与动机

修复 #45588 引入的回归: Gemma4 引擎解析器 (`supports_required_and_named=False`) 下, `tool_choice="required"` 或命名工具选择会返回强制 `FunctionDefinition` JSON 作为 `content`, `tool_calls` 为空。根本原因是基类 `ToolParser.adjust_request` 通过结构化输出约束模型输出 JSON, 但原生 Gemma4 解析器只读取 `<|tool_call>call:... 语法`, 导致 JSON 泄漏到 `content`。

实现拆解

1. 新增 `adjust_request` 覆盖: 在 `vllm/tool_parsers/gemma4_engine_tool_parser.py` 的 `Gemma4EngineToolParser` 类中重写 `adjust_request` 方法, 当 `request.tools` 存在且 `tool_choice` 为 `"required"`、`ChatCompletionNamedToolChoiceParam` 或 `ToolChoiceFunction` 时, 设置 `skip_special_tokens=False` 并直接返回请求, 跳过基类中设置 `structured_outputs` 的逻辑。
2. 导入适配: 添加所需导入 `ChatCompletionRequest`、`ResponsesRequest`、`ChatCompletionNamedToolChoiceParam`、`ToolChoiceFunction`。
3. 新增测试覆盖: 在 `tests/tool_use/test_gemma4_responses_adjust_request.py` 中增加 4 个测试函数, 分别验证 `ChatCompletion` 和 `Responses` 路径下 `required` 和 `named tool choice` 时 `structured_outputs` 被跳过, 同时保留现有 `auto` 测试不变。

关键文件:

- `vllm/tool_parsers/gemma4_engine_tool_parser.py` (模块 工具解析器; 类别 `source`; 类型 `core-logic`; 符号 `adjust_request`): 核心源文件, 新增 `adjust_request` 覆盖, 跳过 `required/named` 的结构化输出约束, 修复回归问题。
- `tests/tool_use/test_gemma4_responses_adjust_request.py` (模块 回归测试; 类别 `test`; 类型 `test-coverage`; 符号 `_build_responses_request`, `_build_chat_request`, `test_gemma4_required_skips_structured_outputs_chatcompletion`, `test_gemma4_named_skips_structured_outputs_chatcompletion`): 新增 4 个测试函数

，覆盖 required/named 在 ChatCompletion 和 Responses 路径下的行为，确保修复有效且不破坏 auto 路径。

关键符号: adjust_request, _build_chat_request,
test_gemma4_required_skips_structured_outputs_chatcompletion

关键源码片段

vllm/tool_parsers/gemma4_engine_tool_parser.py

核心源文件，新增 adjust_request 覆盖，跳过 required/named 的结构化输出约束，修复回归问题。

```
# vllm/tool_parsers/gemma4_engine_tool_parser.py

class Gemma4EngineToolParser(Gemma4ParserToolAdapter):
    supports_required_and_named = False

    def adjust_request(
        self, request: ChatCompletionRequest | ResponsesRequest
    ) -> ChatCompletionRequest | ResponsesRequest:
        """Skip structured-output JSON for required/named tool choice.

        Gemma4 emits its native ``<tool_call>call:...`` syntax, which the
        parser extracts directly. The base ``ToolParser.adjust_request`` would
        set ``structured_outputs`` for required/named and force JSON via guided
        decoding, conflicting with that native syntax (it leaks as content and
        crashes EngineCore under speculative decoding). Skip it so the model
        emits its native format (mirrors the GLM4 parser).
        """

        if request.tools:
            tc = request.tool_choice
            # 当 tool_choice 为 "required" 或命名工具选择时，跳过结构化输出约束
            if tc == "required" or isinstance(
                tc, (ChatCompletionNamedToolChoiceParam, ToolChoiceFunction)
            ):
                request.skip_special_tokens = False
                return request
            # 其他情况（如 auto）继续使用基类逻辑
            return super().adjust_request(request)
```

tests/tool_use/test_gemma4_responses_adjust_request.py

新增 4 个测试函数，覆盖 required/named 在 ChatCompletion 和 Responses 路径下的行为，确保修复有效且不破坏 auto 路径。

```
# tests/tool_use/test_gemma4_responses_adjust_request.py

def test_gemma4_required_skips_structured_outputs_chatcompletion() -> None:
    """required + ChatCompletion: Gemma4EngineToolParser 必须跳过
    强制 JSON structured_outputs，使模型输出原生 <tool_call> 语法。
```

```
基类会约束输出为无法被原生解析器读取的 JSON，导致内容泄漏。"""
parser = Gemma4ToolParser(_StubTokenizer())
request = _build_chat_request(tool_choice="required")

parser.adjust_request(request)

# 验证 structured_outputs 被设置为 None（未触发基类约束）
assert request.structured_outputs is None
# 验证 skip_special_tokens 被设置为 False，保留特殊 token
assert request.skip_special_tokens is False
# 确保工具仍然存在
assert request.tools is not None
```

评论区精华

bbrowning 指出在 DiffusionGemma 合并时已有类似修复，但 rebase 引擎解析器时遗漏。m4r1k 最初尝试在输出端解析 JSON，后同意恢复 adjust_request 方案。双方决定将 required/named JSON 修复与 special tokens 逻辑 (#45553) 分离，保持 PR 聚焦。

- 调整方向：parse JSON vs skip constraint (design): 采用在 Gemma4EngineToolParser 中重写 adjust_request 跳过 JSON 约束的方案。
- PR 范围：只修复 required/named，special tokens 分离 (question): 决定分离，special tokens 问题由 #45553 单独处理。
- 测试验证全面性 (testing): 测试覆盖充分，已通过本地和 live 模型验证。

风险与影响

- 风险：风险较低：调整仅针对 required/named tool choice，通过早返回跳过基类约束，不影响 auto 流程。测试覆盖了 ChatCompletion 和 Responses 两种请求路径。但可能存在未知的引擎解析器交互（如 speculative decoding），已在 live 模型上验证通过。
- 影响：修复影响使用 Gemma4 模型（含 DiffusionGemma）且通过引擎解析器启用 tool-calling 的用户。required 和命名工具选择场景从完全失效变为正常，auto 场景不受影响。
- 风险标记：仅影响 required/named 路径，已在 live 模型验证

关联脉络

- PR #45588 [Frontend] Replace legacy Gemma4 parsers with engine-based implementation: 本 PR 修复了 #45588 引入的回归；#45588 迁移 Gemma4 至引擎解析器时未包含 adjust_request 对 required/named 的处理。
- PR #45553（关联 issue 中提到的 special tokens 讨论，但无具体 PR）：讨论中提及的 special tokens 问题与此 PR 解耦，单独处理。