

PR #48473 完整报告

vllm-project/vllm

[Bugfix] Return 400 instead of 500 when multimodal data is sent to a text-only model

合并时间: 2026-07-14 16:15

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48473>

执行摘要

- 一句话: 修复 text-only 模型收到多模态数据时返回 500 的问题
- 推荐动作: 此 PR 改动简洁, 逻辑清晰, 已通过测试验证, 适合快速合入。值得关注的点: 修复了死代码 (`get_mm_processor()` 中针对 text-only 模型的 `ValueError` 守卫), 以及 reviewer 建议合并测试的良好协作。

功能与动机

Sending multimodal data (e.g. an `image_url`) to a text-only model returns HTTP 500 `InternalServerError`. It is a client mistake and should be HTTP 400 `BadRequest`.

`_mm_req_counter` is only initialized inside the `if mm_registry.supports_multimodal_inputs(...)` block, so on a text-only model the attribute does not exist, causing an `AttributeError` before the existing `ValueError` guard in `get_mm_processor()` can fire.

实现拆解

1. 调整 `_process_multimodal` 方法中的语句顺序 (`vllm/renderers/base.py`): 将 `mm_req_id = f"renderer{self.api_process_rank}-mm-{self._mm_req_counter.inc(1)}"` 从第 738 行移到 `if skip_mm_cache ...` 和 `self.get_mm_processor()` 之后。这样在访问 `_mm_req_counter` 之前, `get_mm_processor()` 先被调用, 对于 text-only 模型会抛出 `ValueError` (原本是死代码), 从而避免 `AttributeError`。
2. 新增 `_build_text_only_renderer` 测试辅助函数 (`tests/renderers/test_process_multimodal_uuids.py`): 使用 `openai-community/gpt2` 作为 text-only 模型构建 `HfRenderer`, 确保模拟真实场景。
3. 添加单个测试方法 `test_text_only_model_mm_data_maps_to_bad_request`: 覆盖两个断言: a) 调用 `_process_multimodal` 应抛出匹配 text-only 的 `ValueError`; b) 通过 `create_error_response` 将该异常转换为 `HTTPStatus.BAD_REQUEST`。
4. 合并原有两个测试为一个 (根据 review 反馈): 最初作者写了两个测试 (一个验证异常类型, 一个验证状态码), reviewer 建议合并, 作者依此修改, 使用 `pytest.raises` 的 `exc_info` 参数同时完成两个断言。

关键文件:

- vllm/renderers/base.py (模块 渲染器; 类别 source; 类型 core-logic; 符号 `_process_multimodal`) : 核心修复文件, 通过调整 `_process_multimodal` 方法中 `mm_req_id` 赋值与 `get_mm_processor()` 调用顺序, 使 text-only 模型异常路径正确触发 `ValueError`。
- tests/renderers/test_process_multi_modal_uuids.py (模块 测试; 类别 test; 类型 test-coverage; 符号 `_build_text_only_renderer`, `test_text_only_model_mm_data_maps_to_bad_request`) : 新增测试覆盖 text-only 模型收到多模态数据的异常路径, 验证返回 400 而非 500。

关键符号: `_process_multimodal`, `_build_text_only_renderer`,
`test_text_only_model_mm_data_maps_to_bad_request`

关键源码片段

vllm/renderers/base.py

核心修复文件, 通过调整 `_process_multimodal` 方法中 `mm_req_id` 赋值与 `get_mm_processor()` 调用顺序, 使 text-only 模型异常路径正确触发 `ValueError`。

```
# vllm/renderers/base.py 中 _process_multimodal 方法的关键片段
# 修改前: mm_req_id = f"renderer{self.api_process_rank}-mm-{self._mm_req_counter.inc(1)}"
# 修改后: 先安全获取 mm_processor, 再使用计数器
def _process_multimodal(
    self,
    prompt: list[int] | str,
    mm_data: MultiModalDataDict,
    mm_uuids: MultiModalUUIDDict | None,
    mm_processor_kwargs: Mapping[str, object] | None,
    tokenization_kwargs: dict[str, Any] | None,
    *,
    skip_mm_cache: bool = False,
) -> "MultiModalInput":
    # Step 1: 解决 multi-modal 处理器 —— 如果模型是 text-only,
    # 这里会抛出 ValueError("Multi-modal processor not available...")
    # 而不是后续访问未初始化的 _mm_req_counter。
    if skip_mm_cache and self._readonly_mm_processor is not None:
        mm_processor = self._readonly_mm_processor
    else:
        mm_processor = self.get_mm_processor()

    # Step 2: 此时模型已确认支持多模态, _mm_req_counter 必然存在
    mm_req_id = f"renderer{self.api_process_rank}-mm-{self._mm_req_counter.inc(1)}"

    mm_data_items = mm_processor.info.parse_mm_data(mm_data)
    mm_uuid_items = parse_mm_uuids(mm_uuids)
    mm_uuid_items = self._process_mm_uuids(
        mm_data, mm_data_items, mm_uuid_items, mm_req_id
    )
    # ... 后续处理不变
```

tests/renderers/test_process_multi_modal_uuids.py

新增测试覆盖 text-only 模型收到多模态数据的异常路径，验证返回 400 而非 500。

```
# tests/renderers/test_process_multi_modal_uuids.py

from http import HTTPStatus
from vllm.entrypoints.serve.utils.error_response import create_error_response

def _build_text_only_renderer() -> HfRenderer:
    """使用 gpt2 作为 text-only 模型构建渲染器。"""
    model_config = ModelConfig(model="openai-community/gpt2", max_model_len=128)
    return HfRenderer(
        VllmConfig(model_config=model_config),
        cached_tokenizer_from_config(model_config),
    )

def test_text_only_model_mm_data_maps_to_bad_request():
    """向 text-only 模型发送多模态数据应返回 400 BadRequest。"""
    renderer = _build_text_only_renderer()

    with pytest.raises(ValueError, match="text-only") as exc_info:
        renderer._process_multimodal(
            prompt="What is in this image?",
            mm_data={"image": [cherry_pil_image]},
            mm_uuids=None,
            mm_processor_kwargs=None,
            tokenization_kwargs=None,
        )

    # 验证异常映射为 400 BadRequest
    error_response = create_error_response(exc_info.value)
    assert error_response.error.code == HTTPStatus.BAD_REQUEST
```

评论区精华

主要讨论来自 reviewer [DarkLight1337](#) 建议合并两个测试为一个，作者 [hnt2601](#) 随后将 [test_text_only_model_rejects_mm_data_with_value_error](#) 和 [test_text_only_model_mm_data_maps_to_bad_request](#) 合并为 [test_text_only_model_mm_data_maps_to_bad_request](#)，因为 `pytest.raises` 可同时匹配异常消息并暴露异常信息，一个测试即可覆盖两个断言。

- 合并测试方法 (testing): 作者 [hnt2601](#) 将两个测试合并为 [test_text_only_model_mm_data_maps_to_bad_request](#)，使用 `pytest.raises` 的 `exc_info` 参数同时验证异常类型与 HTTP 状态码。

风险与影响

- 风险：风险极低：仅调整了两行代码的顺序，且该路径本身就是异常路径（之前返回 500，现在返回 400）。只要 text-only 模型收到多模态数据时 `get_mm_processor()` 正常抛出 `ValueError`，修复就是安全的。所有已有测试均通过，包括 422 媒体错误路径的回归测试。
- 影响：影响范围集中：仅改变 text-only 模型收到多模态数据时的 HTTP 状态码从 500 变为 400。对正常请求路径无影响，对多模态模型无影响。端到端用户感知改善：客户端收到明确的 400 `BadRequest` 而非 500 `InternalServerError`，更容易理解是请求用法错误。
- 风险标记：无

关联脉络

- PR #47595 fix(security): guard lm-format-enforcer regex compile with timeout: 同为前端错误处理改进，修复异常路径返回适当状态码。