

PR #51753 完整报告

vllm-project/vllm

[Misc] Use VLLMValidationError in scoring input validation

合并时间: 2026-08-11 12:12

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/51753>

执行摘要

- 一句话: scoring 输入校验改用 VLLMValidationError, 新增配套测试
- 推荐动作: 值得快速阅读, 作为 RFC #48227 迁移的参考样例。关注点: 后续 VLLMValidationError reparent 时, 需重新审视这些 raise 站点及测试中对 parameter/value 的断言。

功能与动机

RFC #48227 指出 vLLM 入口错误处理碎片化: 代码库依赖原始 Python 异常 (ValueError、TypeError 等), 导致内部错误被误报为 HTTP 400、Prometheus 指标记录错误状态码。本 PR 是该 RFC 迁移的一部分, 将 vllm/entrypoints/pooling/scoring/utils.py 中四个「客户端引起」的校验错误从 ValueError 迁移到 VLLMValidationError: 不支持的 multimodal 输入、输入长度不匹配、空的 query、空的 document。作者在 PR body 中说明, 这些错误共享于 /score、/rerank 与离线 LLM.score(), 因此刻意不设置 parameter/value 字段, 避免绑定单一公共参数名。

实现拆解

1. 新增异常导入: 在 vllm/entrypoints/pooling/scoring/utils.py 头部增加 from vllm.exceptions import VLLMValidationError, 为类型化异常提供依赖。
2. 迁移多模态校验: _validate_mm_score_input 中“非多模态模型收到 MultiModalParam”的分支由 ValueError 改为 VLLMValidationError, 错误消息文本保持不变。
3. 迁移长度校验: _validate_score_input_lens 中三个校验分支 (1:1/1:N/N:N 长度规则、query 为空、document 为空) 同步迁移, 保持消息不变。
4. 保留结构化字段: 不填充 parameter/value 字段, 因为同一工具函数被多个入口 (HTTP 与离线 API) 共用, 公共参数名不一致, 避免绑定单一字段名。
5. 测试配套: 新增 tests/entrypoints/pooling/scoring/test_utils.py, 用 pytest.mark.parametrize 覆盖四种错误路径, 断言异常类型、消息文本及 parameter/value 为 None。本地测试结果 4 passed, pre-commit 全部通过。

关键文件:

- vllm/entrypoints/pooling/scoring/utils.py (模块 评分校验; 类别 source; 类型 core-logic; 符号 _validate_mm_score_input, _validate_score_input_lens, validate_score_input): 核心变更文件, 将四处客户输入校验异常从 ValueError 迁移到 VLLMValidationError, 并

新增异常导入。

- tests/entrypoints/pooling/scoring/test_utils.py (模块 评分测试; 类别 test; 类型 test-coverage; 符号 test_validate_score_input_rejects_invalid_inputs) : 新增参数化测试覆盖四种异常场景, 断言异常类型、消息与 parameter/value 字段。

关键符号: _validate_mm_score_input, _validate_score_input_lens, validate_score_input, test_validate_score_input_rejects_invalid_inputs

关键源码片段

vllm/entrypoints/pooling/scoring/utils.py

核心变更文件, 将四处客户输入校验异常从 ValueError 迁移到 VLLMValidationError, 并新增异常导入。

将每条 score 输入做多模态兼容性检查, 返回规范化的 ScoreData 列表

```
def _validate_mm_score_input(
    data: list[ScoreInput],
    is_multimodal_model: bool,
    architecture: str,
) -> list[ScoreData]:
    out: list[ScoreData] = []
    for d in data:
        if isinstance(d, str):
            out.append(d)
        else:
            # 非多模态模型收到 MultiModalParam 时抛类型化校验异常
            if not is_multimodal_model:
                raise VLLMValidationError(
                    f"MultiModalParam is not supported for {architecture}"
                )
            content = cast(list[ScoreContentPartParam], d.get("content", []))
            out.append(content)
    return out
```

```
def _validate_score_input_lens(
    data_1: list[ScoreData],
    data_2: list[ScoreData],
):
    len_1 = len(data_1)
    len_2 = len(data_2)

    # 只允许 1:1、1:N、N:N 三种长度组合, 否则抛错
    if len_1 > 1 and len_1 != len_2:
        raise VLLMValidationError("Input lengths must be either 1:1, 1:N or N:N")
    # query 与 document 均不允许为空
    if len_1 == 0:
        raise VLLMValidationError("At least one text element must be given")
```

```

if len_2 == 0:
    raise VLLMValidationError("At least one text_pair element must be given")

def validate_score_input(
    data_1: ScoreInput | list[ScoreInput],
    data_2: ScoreInput | list[ScoreInput],
    is_multimodal_model: bool,
    architecture: str,
) -> ScoringData:
    # 统一入口：先规范化单输入为列表，再依次做多模态与长度校验
    if not isinstance(data_1, list):
        data_1 = [data_1]

    if not isinstance(data_2, list):
        data_2 = [data_2]

    score_input_1 = _validate_mm_score_input(data_1, is_multimodal_model, architecture)
    score_input_2 = _validate_mm_score_input(data_2, is_multimodal_model, architecture)
    _validate_score_input_lens(score_input_1, score_input_2)
    return ScoringData(data_1=score_input_1, data_2=score_input_2)

```

tests/entrypoints/pooling/scoring/test_utils.py

新增参数化测试覆盖四种异常场景，断言异常类型、消息与 `parameter/value` 字段。

参数化覆盖四种客户端错误：非法多模态输入、长度不匹配、query 为空、document 为空

```

@pytest.mark.parametrize(
    ("data_1", "data_2", "is_multimodal_model", "architecture", "message"),
    [
        pytest.param(
            {"content": []},
            "document",
            False,
            "TestModel",
            "MultiModalParam is not supported for TestModel",
            id="unsupported-multimodal-input",
        ),
        pytest.param(
            ["query 1", "query 2"],
            ["document"],
            False,
            "TestModel",
            "Input lengths must be either 1:1, 1:N or N:N",
            id="incompatible-input-lengths",
        ),
        pytest.param(
            [],
            ["document"],
            False,

```

```

        "TestModel",
        "At least one text element must be given",
        id="empty-first-input",
    ),
    pytest.param(
        ["query"],
        [],
        False,
        "TestModel",
        "At least one text_pair element must be given",
        id="empty-second-input",
    ),
],
)
def test_validate_score_input_rejects_invalid_inputs(
    data_1: ScoreInput | list[ScoreInput],
    data_2: ScoreInput | list[ScoreInput],
    is_multimodal_model: bool,
    architecture: str,
    message: str,
):
    with pytest.raises(VLLMValidationError) as exc_info:
        validate_score_input(
            data_1,
            data_2,
            is_multimodal_model=is_multimodal_model,
            architecture=architecture,
        )

# VLLMValidationError 保留 parameter/value 字段为空，等待后续 RFC 细化
assert str(exc_info.value) == message
assert exc_info.value.parameter is None
assert exc_info.value.value is None

```

评论区精华

该 PR 来自 fork, [claude\[bot\]](#) 的自动化 review 被禁用, 维护者 [noooop](#) 触发 Buildkite CI #83282 后批准合并并致谢 (“thanks!”)。PR 本身没有实质性技术争论; 作者在 body 中声明已搜索 #48227 相关 PR 并检查当前 main 实现, 确认没有重复的迁移 PR, 并在 RFC 内讨论过本文件级范围。

- fork 仓库无法自动 review (other): 维护者 noooop 手动触发 CI 并批准合并。

风险与影响

- 风险:

1. 兼容性风险: VLLMValidationError 目前仍继承自 ValueError (RFC 计划后续 reparent 到 VLLMClientError), 因此现有 except ValueError 的调用方仍能捕获; 但

RFC 完成后继承链改变，届时依赖 ValueError 的捕获可能失效，属中间态风险。

2. 行为一致性：错误消息与 HTTP 400 映射均未改变，在线服务无感知；离线 LLM.score() 调用者会从 ValueError 感知为 VLLMValidationError，属于预期内的类型升级。
3. 测试覆盖：新增测试覆盖四种校验失败路径，但没有覆盖在线端点端到端映射，回归风险低。
4. 影响范围：仅影响 /score、/rerank 输入校验路径，不影响模型执行与采样逻辑。
 - 影响：对用户而言，错误消息保持不变，在线 /score、/rerank 的 HTTP 400 映射不变；离线 LLM.score() 调用者获得更精确的异常类型，便于区分客户端错误与服务端错误。对团队而言，这是 RFC #48227 标准化错误处理的一步小规模落地，为后续 VLLMValidationError 从 ValueError 解耦提供了可参照的迁移模式（窄范围、保持消息稳定、测试配套）。 - 风险标记：离线调用异常类型变更，VLLMValidationError 继承链将变

关联脉络

- PR #48227 [RFC]: Standardize vLLM Entrypoint Error Handling: 本 PR 是该 RFC 的具体实施步骤之一，四个 ValueError → VLLMValidationError 的迁移对应 RFC 的 Step 2/3。
- PR #51654 Fix chat completion 500 on non-object JSON bodies: 同为前端入口错误处理修复，将错误分类与 HTTP 状态码映射对齐，属于同一错误处理标准化方向。