PR #51795 完整报告

vllm-project/vllm

[Bugfix] Reject negative token ids as out-of-vocabulary

合并时间：2026-08-14 09:07

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51795

执行摘要

- 一句话：在校验路径上拒绝负 token id，返回 HTTP 400
- 推荐动作：值得精读的输入校验鲁棒性修复。重点关注两点：一是共享校验路径上如何以最小成本补齐边界检查；二是测试位置的调整（从 completion 到 pooling）体现了对 OpenAI schema 与引擎级校验分层拦截的理解，对类似校验场景有借鉴意义。

功能与动机

调用方传入 token id 后，该 id 会在下游作为 embedding 索引使用，负索引永远无效；但现有校验只检查上界，导致负 id 通过校验并最终触发 CUDA 设备端断言使服务崩溃（ECMGit 在评论中确认此前 e2e 场景下服务器会崩溃且 PyTorch 无法捕获该输入）。PR 目的即补上对称的下界检查，在共享输入校验路径上统一拦截非法 token id。

实现拆解

1. 修改校验入口：在 vllm/v1/engine/input_processor.py 的 _validate_model_input 方法中，新增 min_input_id = min(prompt_ids, default=0)，并在计算 model_vocab_size 后增加 if min_input_id < 0: raise VLLMValidationError(...)。该错误经 VLLMValidationError 映射为 HTTP 400。
2. 保持上界逻辑不变：原有的 max_input_id > max(tokenizer.max_token_id, model_vocab_size - 1) 检查继续保留，新增检查与之对称，不改变已有合法上限的判定。
3. 补充回归测试：在 tests/entrypoints/pooling/embed/test_online.py 中新增 test_negative_token_ids，通过 /v1/embeddings 接口发送 input=[-1]，断言返回 openai.BadRequestError 且消息匹配 "out of vocabulary"。测试放在 pooling 端点是因为 OpenAI completion schema 已用 Field(ge=0) 在 pydantic 层拦截负 id，无法到达引擎层校验，而 pooling schema 没有该约束（见 commit 8e6a6d0）。

关键文件：

- vllm/v1/engine/input_processor.py（模块 输入校验；类别 source；类型 core-logic；符号 _validate_model_input）：核心修复文件：在共享输入校验路径 _validate_model_input 中新增负 token id 下界检查，是该 PR 的行为变更主体。
- tests/entrypoints/pooling/embed/test_online.py（模块 嵌入端点；类别 test；类型 test-coverage；符号 test_negative_token_ids）：回归测试：验证负 token id 在引擎级共享校验路径上被拒绝，并说明为何选择 pooling 端点而非 completion 端点。

关键符号：_validate_model_input

关键源码片段

vllm/v1/engine/input_processor.py

核心修复文件：在共享输入校验路径 _validate_model_input 中新增负 token id 下界检查，是该 PR 的行为变更主体。

```
# vllm/v1/engine/input_processor.py（节选：_validate_model_input 中的 vocab 校验部分）
if prompt_ids and tokenizer is not None:
    max_input_id = max(prompt_ids, default=0)
    min_input_id = min(prompt_ids, default=0) # 新增：对称的下界计算

    # 这里取 tokenizer 与 model 两者 vocab 的较大者判断上界，
    # 因为 Qwen3 等模型的 tokenizer 与 model vocab 大小可能不一致。
    model_vocab_size = model_config.get_vocab_size()

    # 负 token id 同样属于 out-of-vocabulary，但原上界检查无法捕获；
    # 若放任其进入下游，会以负数索引 embedding，触发 CUDA 设备端断言。
    # 该路径为 generate / embedding / pooling 共享，因此一处修复覆盖三类请求。
    if min_input_id < 0:
        raise VLLMValidationError(
            f"Token id {min_input_id} is out of vocabulary"
        )
    if max_input_id > max(tokenizer.max_token_id, model_vocab_size - 1):
        raise VLLMValidationError(
            f"Token id {max_input_id} is out of vocabulary"
        )
```

tests/entrypoints/pooling/embed/test_online.py

回归测试：验证负 token id 在引擎级共享校验路径上被拒绝，并说明为何选择 pooling 端点而非 completion 端点。

```
# tests/entrypoints/pooling/embed/test_online.py（新增测试）
@pytest.mark.asyncio
@pytest.mark.parametrize("model_name", [MODEL_NAME])
async def test_negative_token_ids(client: openai.AsyncOpenAI, model_name: str):
    # 完成端点（/v1/completions）的 schema 已用 Field(ge=0) 拒绝负 token id，
    # 请求到不了引擎层；而 pooling 的 embeddings schema 没有该约束，
    # 因此这里通过 /v1/embeddings 验证共享的引擎级校验路径是否兜底。
    with pytest.raises(openai.BadRequestError, match=".*out of vocabulary.*"):
        await client.embeddings.create(model=model_name, input=[-1])
```

评论区精华

- DarkLight1337 在 issue 评论中询问 *e2e* 场景下传负 token id 会发生什么，ECMGit 回答会导致 CUDA 设备端断言崩溃，且 PyTorch 无法提前捕获，这直接论证了本次修复的必要性。

- njhill 在 review 中对 min() 与 max() 两次遍历长 prompt 提出性能疑问，随后自我否定（"built-in functions are faster"），最终未要求改成单循环，确认该实现性能无损。
- e2e 场景下负 token id 是否会导致崩溃 (question): 确认负 token id 是真实崩溃源，验证了修复必要性。
- 长 prompt 下 min/max 两次遍历的性能 (performance): 接受内置 min/max 实现，无需优化。

风险与影响

- 风险：
  - 共享路径影响面：校验位于 generate / embedding / pooling 三类请求共用的 _validate_model_input，行为收紧会同时影响所有入口，需回归这三类请求的正常输入路径。
  - 隐式行为变更：此前负 token id 可能以 Python 负索引语义流转到深层（尽管最终会崩溃或产生错误结果），现在直接返回 400，属于破坏性但正确的行为变化，需确认没有用户依赖该异常行为。
  - 测试覆盖盲区：新增测试只覆盖了 pooling 端点，completion 端点因 pydantic 拦截（错误消息为字段约束而非 "out of vocabulary"）未在本 PR 内验证，存在两类端点错误消息不一致的轻微文档 / 行为差异。
- 影响：
  - 用户侧：传入负 token id 的请求从服务崩溃变为干净的 HTTP 400 错误，显著提升错误可诊断性。
  - 系统侧：避免 CUDA 设备端断言导致的进程级崩溃，提高服务稳定性；新增一个 min() 遍历，对长 prompt 的性能影响可忽略（njhill 已确认内置函数更快）。
  - 团队侧：改动仅 21 行（源码 10 行、测试 11 行），风险低，维护成本小；为后续更多输入边界校验提供了可参考的对称检查模式。
  - 风险标记：共享校验路径变更，行为收紧，测试仅覆盖 pooling 端点

关联脉络

- PR #51796 [Bugfix] Reject NUL byte in structured_outputs.regex: 同为输入校验类 bugfix，在 v1 前端 / 校验路径上拒绝非法输入并带回退测试，与该 PR 的边界校验思路一致。