PR #28088 完整报告

sgl-project/sglang

fix(frontend): return HTTP 400 for out-of-vocabulary token_ids_logprob

合并时间：2026-06-13 08:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/28088

执行摘要

- 一句话：修复 token_ids_logprob 越界导致引擎崩溃
- 推荐动作：值得精读，尤其是想了解如何在请求处理早期注入输入验证以预防 GPU 崩溃的工程师。新增的 _validate_token_ids_logprob 方法和对应的测试用例可作为类似的输入校验参考。

功能与动机

/generate 接受用户提供的 token_ids_logprob，但 TokenizerManager._validate_one_request 未校验这些 ID。越界 ID（>= vocab_size）导致 CUDA device-side assert 并杀死引擎，后续合法请求也会失败；负数 ID 被 PyTorch 索引接受并返回错误 token 的 logprob。这是 vLLM 已修复的同一类输入问题（vllm-project/vllm#44042）。

实现拆解

1. 在 TokenizerManager 新增 _validate_token_ids_logprob 方法（python/sglang/srt/managers/tokenizer_manager.py）：
   - 检查 token_ids_logprob 是否为非空列表，否则直接返回（空或 None 视为无需校验）。
   - 遍历每个元素，若不为 int 则抛出带有明确提示的 ValueError。
   - 若元素 < 0 或 >= self.model_config.vocab_size，抛出 ValueError，错误信息包含具体 ID 和合法范围。
2. 在 _validate_one_request 的 GenerateReqInput 分支中调用新方法：
   - 位置放在 return_hidden_states 和 custom_logit_processor 校验之前，确保在最早入口拦截非法输入。
   - 即使 return_logprob=false 也执行校验（与 vLLM 行为一致）。
3. 在测试文件中新增 4 个测试用例（test/registered/openai_server/validation/test_request_length_validation.py）：
   - test_token_ids_logprob_out_of_vocabulary：验证负数和超大 ID 均返回 HTTP 400 且错误消息包含 "out-of-vocabulary"。
   - test_token_ids_logprob_rejects_nested_list：验证嵌套列表（即使 ID 合法）被拒绝，返回 HTTP 400 且包含 "flat list of integers"。

- test_token_ids_logprob_batch_with_one_oov：验证批量请求中一个子请求包含越界 ID 时返回 HTTP 400。
- test_token_ids_logprob_valid：验证合法 ID 返回 HTTP 200。

关键文件：

- python/sglang/srt/managers/tokenizer_manager.py（模块 请求处理；类别 source；类型 core-logic；符号 _validate_token_ids_logprob）：核心变更文件，新增 _validate_token_ids_logprob 方法并在 _validate_one_request 中调用，是修复逻辑的入口。
- test/registered/openai_server/validation/test_request_length_validation.py（模块 测试；类别 test；类型 test-coverage；符号 test_token_ids_logprob_out_of_vocabulary, test_token_ids_logprob_rejects_nested_list, test_token_ids_logprob_batch_with_one_oov, test_token_ids_logprob_valid）：新增 4 个测试用例覆盖了外词汇、嵌套列表、批量混合和合法请求，验证修复的有效性。

关键符号：_validate_token_ids_logprob

关键源码片段

test/registered/openai_server/validation/test_request_length_validation.py

新增 4 个测试用例覆盖了外词汇、嵌套列表、批量混合和合法请求，验证修复的有效性。

```
def test_token_ids_logprob_out_of_vocabulary(self):
    headers = {"Authorization": f"Bearer {self.api_key}"}
    for token_ids_logprob in ([-1], [2_000_000_000]):
        response = requests.post(
            f"{self.base_url}/generate",
            headers=headers,
            json={
                "text": "hi",
                "sampling_params": {"max_new_tokens": 1},
                "return_logprob": True,
                "token_ids_logprob": token_ids_logprob,
            },
        )
        self.assertEqual(response.status_code, 400)
        self.assertIn("out-of-vocabulary", response.text)

def test_token_ids_logprob_rejects_nested_list(self):
    # 嵌套列表不是单请求应有的格式；不规则嵌套即使 ID 合法也会导致 GPU 崩溃
    headers = {"Authorization": f"Bearer {self.api_key}"}
    for token_ids_logprob in ([[0]], [[0], [1, 2]]):
        response = requests.post(
            f"{self.base_url}/generate",
            headers=headers,
            json={
                "text": "hi",
                "sampling_params": {"max_new_tokens": 1},
```

```
                "return_logprob": True,
                "token_ids_logprob": token_ids_logprob,
            },
        )
        self.assertEqual(response.status_code, 400)
        self.assertIn("flat list of integers", response.text)

    def test_token_ids_logprob_batch_with_one_oov(self):
        # 批量请求中一个子请求越界，整体应被拒绝
        headers = {"Authorization": f"Bearer {self.api_key}"}
        response = requests.post(
            f"{self.base_url}/generate",
            headers=headers,
            json={
                "text": ["hi", "hi"],
                "sampling_params": {"max_new_tokens": 1},
                "return_logprob": True,
                "token_ids_logprob": [[0], [2_000_000_000]],
            },
        )
        self.assertEqual(response.status_code, 400)
        self.assertIn("out-of-vocabulary", response.text)

    def test_token_ids_logprob_valid(self):
        headers = {"Authorization": f"Bearer {self.api_key}"}
        response = requests.post(
            f"{self.base_url}/generate",
            headers=headers,
            json={
                "text": "hi",
                "sampling_params": {"max_new_tokens": 1},
                "return_logprob": True,
                "token_ids_logprob": [0],
            },
        )
        self.assertEqual(response.status_code, 200)
```

评论区精华

- hnyls2002 指出了嵌套列表的额外风险：Review 评论指出，批量请求已被拆分为单请求后，嵌套 token_ids_logprob（如 [[0]] 或 [[0], [1, 2]]）只能来自格式错误的输入。一个内部 ID 合法但形状不规则的嵌套列表（如 [[0], [1, 2]]）会通过初始的 OOV 校验，但随后在采样器的 torch.tensor(...) 上因不规则列表引发崩溃——这正是本 PR 要修复的同一类故障。hnyls2002 随后推送了第二个 commit 来扩展验证，增加了对非列表类型和非整型元素的检查，以及相应的测试用例。
- 设计权衡：本 PR 选择在 TokenizerManager 层做输入验证（而非更底层），因为此时 vocab_size 已知且能返回 HTTP 400 错误码，避免了 GPU 段错误。

- 嵌套列表的风险和验证增强 (correctness): 作者接受建议并推送第二个 commit 追加了非列表和非整型校验，同时增加嵌套列表拒绝测试。

风险与影响

- 风险：
  - 低风险：变更集中在输入验证路径，不影响正常请求的推理逻辑。仅当 token_ids_logprob 非法时才返回 400，合法请求路径完全不变。
  - 测试覆盖充分：负面用例（负数、超大、嵌套列表、批量混合）和正面用例均覆盖，且已在多模型（TinyLlama、Qwen3 系列、gpt-oss-20b 等）上运行 11 项矩阵测试全部通过。
  - 兼容性风险：已隐式要求 token_ids_logprob 为 list[int]，之前可能被容忍的格式（如元组、numpy 数组）现在会报 400，但实际使用中罕见，可视为安全性改进。
- 影响：
  - 用户影响：通过 /generate 传入非法 token_ids_logprob 的用户将立即收到 HTTP 400 错误及明确提示（如 "out-of-vocabulary token id 999999999; valid range is [0, 151936)."），而非服务端崩溃或错误输出。保护了后续请求不被牵连。
  - 系统影响：消除了由该字段引起的 CUDA device-side assert 崩溃，提升了引擎稳定性。
  - 团队影响：新增的验证方法与已有的 _validate_input_ids_in_vocab 模式一致，易于维护。
  - 风险标记：输入校验，引擎稳定性

关联脉络

- 暂无明显关联 PR