PR #46348 完整报告

vllm-project/vllm

[Rust Frontend] Align Rust allowed_token_ids validation with Python

合并时间：2026-06-23 16:32

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46348

执行摘要

此 PR 修复 Rust OpenAI 前端与 Python 端在 allowed_token_ids 参数上的行为不一致：当显式传入空列表 [] 时，Python 端会拒绝并报错，而 Rust 端之前仅检查 out-of-vocab 情况，导致空列表被接受。现在通过在 Rust 请求验证路径中添加 EmptyAllowedTokenIds 错误变体，两端行为完全对齐，返回相同的 400 invalid_request_error。

功能与动机

Python 端的 SamplingParams 会检查 allowed_token_ids is not None and empty! 并拒绝空列表，但 Rust 前端仅校验 token ID 是否超出词表，忽略空列表场景。为了消除跨前端的行为歧义，PR 为其补上了这一校验，确保客户端无论使用 Python 还是 Rust 的前端入口，都能收到一致的报错提示。

实现拆解

1. 错误类型重构（rust/src/text/src/lower/token_ids.rs）：将原有的 OutOfVocabError 结构体重构成 TokenIdsError 枚举，新增 EmptyAllowedTokenIds 变体，并把原结构体作为 OutOfVocab 枚举变体内联，保持错误类型层次单一。
2. 核心验证函数增强（rust/src/text/src/lower/token_ids.rs）：在 validate_vocab_range 中，当 allowed_token_ids 存在且为空时，立即返回 EmptyAllowedTokenIds 错误，不再继续校验词表范围。
3. 测试用例更新（rust/src/text/src/lower.rs）：将原有所有匹配 OutOfVocabError 的断言替换为匹配 TokenIdsError::OutOfVocab，并新增 lower_sampling_params_rejects_empty_allowed_token_ids 单元测试，验证空列表被正确拒绝。
4. 错误映射与导出：
   - rust/src/text/src/error.rs & rust/src/text/src/lib.rs：替换 pub use 和 Error 变体，将 OutOfVocabError 改为 TokenIdsError。
   - rust/src/server/src/error.rs：更新 is_request_validation_error 函数匹配的变体，并新增 empty_allowed_token_ids_maps_to_invalid_request 单元测试，确保新错误映射到 HTTP 400。
5. 集成测试覆盖（rust/src/server/src/routes/tests.rs）：新增两个 e2e 测试，分别对 /v1/completions 和 /v1/chat/completions 传入空 allowed_token_ids，验证返回 400 状态码及正确的错误消息。

rust/src/text/src/lower/token_ids.rs

核心验证逻辑变更：引入 TokenIdsError 枚举并添加空列表检查。

```
// 文件：rust/src/text/src/lower/token_ids.rs
// 引入的 TokenIdsError 枚举，统一 token ID 验证错误
#[derive(Debug, Error)]
pub enum TokenIdsError {
    // 新增：allowed_token_ids 为空时使用此变体
    #[error("`allowed_token_ids` should not be empty")]
    EmptyAllowedTokenIds,
    // 原有 out-of-vocab 错误内联为枚举变体
    #[error(
        "token_id(s) {token_ids:?} in {parameter} contain out-of-vocab token ids. \
         Vocabulary size: {vocab_size}"
    )]
    OutOfVocab {
        parameter: &'static str,
        token_ids: Vec<u32>,
        vocab_size: usize,
    },
}

// validate_vocab_range 函数：校验采样参数中的 token ID 范围
pub(crate) fn validate_vocab_range(
    params: &EngineCoreSamplingParams,
    limits: &SamplingLimits,
) -> Result<(), TokenIdsError> {
    // 校验 stop_token_ids
    validate_param("stop_token_ids", params.stop_token_ids.iter().copied(), limits.model_vocab_
    size)?;

    // 校验 allowed_token_ids：先检查是否为空，再校验范围
    if let Some(token_ids) = params.allowed_token_ids.as_deref() {
        // 新增空列表检查：Python 端同样拒绝此情况
        if token_ids.is_empty() {
            return Err(TokenIdsError::EmptyAllowedTokenIds);
        }
        validate_param("allowed_token_ids", token_ids.iter().copied(), limits.tokenizer_vocab_size)?;
    }

    // 其余 logit_bias、logprob_token_ids、bad_words_token_ids 校验保持不变
    // ...
    Ok(())
}
```

rust/src/server/src/error.rs

更新错误映射和测试，确保新错误类型映射到 HTTP 400。

```
// 文件：rust/src/server/src/error.rs
// 判断是否为请求验证错误的函数
fn is_request_validation_error(error: &vllm_text::Error) -> bool {
```

```
    matches!(
        error,
        vllm_text::Error::PromptTooLong { .. }
            | vllm_text::Error::EmptyPromptTokenIds { .. }
            | vllm_text::Error::Logprobs(_)
            // 将旧的 OutOfVocab 变体替换为新的 TokenIds 变体
            | vllm_text::Error::TokenIds(_)
            | vllm_text::Error::InvalidThinkingTokenBudget
            | vllm_text::Error::Llm(vllm_llm::Error::EmptyPromptTokenIds { .. })
    )
}

// 新增测试：验证空 allowed_token_ids 映射为 400 invalid_request_error
#[test]
fn empty_allowed_token_ids_maps_to_invalid_request() {
    let error = vllm_text::Error::TokenIds(vllm_text::TokenIdsError::EmptyAllowedTokenIds);
    let api_error = text_submit_error("failed to submit completion request", error);
    assert_eq!(api_error.status_code(), StatusCode::BAD_REQUEST);
    let response = api_error.to_error_response();
    assert_eq!(response.error.error_type, "invalid_request_error");
    assert!(response.error.message.contains("allowed_token_ids"));
}
```

评论区精华

- BugenZhao建议将独立的 OutOfVocabError 结构体内联为 TokenIdsError 枚举变体，避免 text::Error 中保留两个独立变体。此重构被作者采纳，最终提交中 OutOfVocabError 成为 TokenIdsError::OutOfVocab。
- BugenZhao还建议调整错误消息格式，将 "allowed_token_ids is not None and empty!" 改为 "allowed_token_ids should not be empty"，使信息更简洁且与项目风格一致。最终提交已更新。

风险与影响

- 向后兼容性：之前能通过的 allowed_token_ids: [] 请求现在会被拒绝。这属于修复行为不一致，不算是兼容性 break，但客户端如果存在此类调用需要更新。
- 测试覆盖：单元测试和集成测试均已补充，风险很低。
- 影响范围：仅影响请求验证早期阶段，不涉及引擎核心或性能路径。

关联脉络

此 PR 是 Rust 前端参数验证体系持续完善的一部分。与之关联的 #PR 46359 同样修改了 lower.rs 和 error.rs，修正了 --reasoning-parser 的语义。两者共同提升了 Rust 前端与 Python 前端的验证一致性。