

PR #47126 完整报告

vllm-project/vllm

[Bugfix] Fix beam search candidate indexing when logprobs count varies

合并时间: 2026-07-01 15:07

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47126>

执行摘要

- 一句话: 修复 beam search 中 logprobs 数量不同时的索引越界
- 推荐动作: 建议合并。该 PR 修复了真实 bug, 重构后逻辑更直观易维护, 且通过了已有和新增测试。

功能与动机

CI 集成测试中遇到 `IndexError: list index out of range`, 具体发生在 `vllm/entrypoints/generate/beam_search/online.py:150: current_beam = all_beams[idx // logprobs_num]`。根本原因是当不同 beam 的 logprobs 数量不同时, `logprobs_num` 不是恒定值, 导致错误的 beam 索引。此 bug 由畸形的输入触发。

实现拆解

1. 移除全局索引数组: 在 `beam_search` 主循环中, 删除原有的 `all_beams_token_id` 和 `all_beams_logprob` 列表, 改为在每个 beam 推理结果循环中直接构建 `candidates` 列表。
2. 逐 token 处理 logprobs: 对于每个 beam 的 logprobs (`result.outputs[0].logprobs[0]`), 遍历其所有 token, 若为 EOS 且未忽略则直接加入 `completed` 列表; 否则将 `(cum_logprob, token_id, beam, logprobs)` 元组追加到 `candidates`。
3. 候选排序与 Top-K 选择: 使用 `np.fromiter` 收集所有候选的 `cum_logprob`, 根据候选数量与 `beam_width` 的关系选择 `argsort` 或 `argpartition` (部分排序后再排序以保证稳定性), 得到 top-K 索引。
4. 构建新 beams: 遍历 top-K 索引, 从 `candidates` 中取出对应元组并创建 `BeamSearchSequence` 实例作为下一轮 beams。
5. 新增测试: 测试文件 `tests/samplers/test_beam_search_online.py` 模拟 `beam_width=2` 且 logprobs 包含额外候选 token 的场景, 验证最终选中 EOS。

关键文件:

- `vllm/entrypoints/generate/beam_search/online.py` (模块 束搜索; 类别 source; 类型 core-logic): 核心修复文件, 重构 beam search 候选索引逻辑
- `tests/samplers/test_beam_search_online.py` (模块 束搜索; 类别 test; 类型 test-coverage; 符号 `_Tokenizer`, `decode`, `_Renderer`, `get_tokenizer`): 新增测试用例, 覆盖 logprobs 数量不同时 beam search 的行为

关键符号: beam_search, test_beam_search_handles_extra_logprob_candidates

关键源码片段

[vllm/entrypoints/generate/beam_search/online.py](#)

核心修复文件, 重构 beam search 候选索引逻辑

```
# beam_search 方法主循环内 (for _ in range(max_tokens))
candidates = []
for i, result in enumerate(output):
    current_beam = all_beams[i]
    # 检查错误 finish reason
    if result.outputs[0].finish_reason == "error":
        yield RequestOutput(..., finished=True)
        return
    if result.outputs[0].logprobs is not None:
        logprobs = result.outputs[0].logprobs[0]
        for token_id, logprob_obj in logprobs.items():
            candidate_logprob = current_beam.cum_logprob + logprob_obj.logprob
            if token_id == eos_token_id and not ignore_eos:
                completed.append(BeamSearchSequence(...))
            else:
                candidates.append((candidate_logprob, int(token_id), current_beam, logprobs))

# 从 candidates 提取对数概率数组
candidate_logprobs = np.fromiter(
    (c[0] for c in candidates), dtype=np.float64, count=len(candidates))
# 选择 top-K 索引
if len(candidates) <= beam_width:
    topn_idx = np.argsort(-candidate_logprobs)
else:
    topn_idx = np.argpartition(-candidate_logprobs, beam_width - 1)[:beam_width]
    topn_idx = topn_idx[np.argsort(-candidate_logprobs[topn_idx])]

# 构建 new_beams
new_beams = []
for idx in topn_idx:
    cum_logprob, token_id, current_beam, logprobs = candidates[int(idx)]
    new_beams.append(BeamSearchSequence(
        orig_prompt=prompt,
        tokens=current_beam.tokens + [token_id],
        logprobs=current_beam.logprobs + [logprobs],
        lora_request=current_beam.lora_request,
        cum_logprob=cum_logprob,
    ))
all_beams = new_beams
if not all_beams:
    break
```

评论区精华

无实质 review 讨论，仅两个批准。reviewer yzong-rh 表示本地无法复现 `IndexError`，但审查后认为新代码逻辑正确且测试通过。

- 代码正确性确认 (correctness): 批准合并

风险与影响

- 风险：风险较低，因为本质是同一逻辑的简化重构，且新增了专门的测试覆盖。主要风险是被重构部分对性能的影响：使用 Python 层循环替代了 NumPy 批量操作，但在 beams 数量较小（通常 ≤ 10 ）时可忽略。
- 影响：影响使用 beam search 的 API 用户（v1/completions 和 v1/chat/completions 中 `use_beam_search=True`）。修复了当 logprobs 数量变化时返回 500 错误的 bug，提升了 beam search 的鲁棒性。
- 风险标记：核心索引逻辑重构，依赖测试覆盖，beam search 路径变更

关联脉络

- 暂无明显关联 PR