

PR #56505 完整报告

vllm-project/vllm

[Pooling] Cap max-length padding for chunked embeddings

合并时间: 2026-09-16 08:31

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56505>

执行摘要

- 一句话: chunked 嵌入 padding 上限改为 $\min(\text{max_model_len}, \text{max_embed_len})$
- 推荐动作: 该 PR 值得快速精读, 尤其关注其设计取舍: 以 $\min(\text{max_model_len}, \text{max_embed_len})$ 作为填充封顶, 既修复了短输入浪费, 又保留 max_embed_len 作为真实长输入的验证上限, 是典型的“保留校验边界、修正实际填充目标”的做法。对 pooling 服务维护者来说, 评论中关于“先 padding 还是先 chunk”的权衡也值得吸收: 避免 pad-only chunk 污染加权 embedding 是设计关键。建议后续补充一条针对该组合路径的自动化测试, 防止回归。

功能与动机

PR body 明确指出: “Padding incorrectly used the aggregate max_embed_len, creating artificial pad-only chunks.” 复现中 hello 被填充到 prompt_tokens=10000, 而修复后为 512。作者在评论中总结: 减少计算与内存、token 计数正确、避免 pad-only chunk 污染加权嵌入, 同时真实长输入仍可支持 10000 token。

实现拆解

实现拆解如下:

1. 变更入口: 在 `vllm/entrypoints/pooling/base/protocol.py` 的 `EmbeddingTokenizeParamsMixin` 中新增 `padding: Literal["max_length", "do_not_pad"] | None` 字段, 用于读取请求中的 padding 参数, 使 mixin 内部能够感知当前请求是否要求 max_length 填充。
2. 重构返回路径: 将 `build_tok_params` 原来直接 `return self._build_pooling_tok_params(...)` 的写法改为先构造 `tok_params`, 再根据条件叠加 padding 覆盖。这是为后续按需追加 `pad_prompt_tokens` 参数做准备, 不改变非 chunked 路径的行为。
3. 核心修正逻辑: 当 `self.padding == "max_length"`、`pooler_config` 存在且 `enable_chunked_processing` 为真时, 计算 `max_padding_len` (优先取 `pooler_config.max_embed_len`, 为 `None` 时回退到 `max_model_len`), 并通过 `tok_params.with_kwargs(pad_prompt_tokens=min(default_max_total_tokens, max_padding_len))` 将填充目标封顶。这样既避免按 10000 填充, 又保留用户显式配置的更小 embedding 上限。

4. 测试与配套：本次未新增或修改测试文件，仅依赖现有 `tests/entrypoints/pooling/embed/test_protocol.py`、`test_io_processor.py` (66 passed) 和 `test_online_long_text.py` (5 passed) 验证；作者另在 H100 上做了单 token、513 token 边界、批量混合等多组数值对照实验，确认余弦相似度为 1.0。

关键文件：

- `vllm/entrypoints/pooling/base/protocol.py` (模块 池化入口；类别 source；类型 core-logic；符号 `EmbeddingTokenizeParamsMixin.padding`, `EmbeddingTokenizeParamsMixin.build_tok_params`)：唯一变更文件，包含全部修复逻辑：新增 `padding` 字段、重构 `build_tok_params` 返回路径、追加 `pad_prompt_tokens` 封顶。

关键符号：`EmbeddingTokenizeParamsMixin.build_tok_params`

关键源码片段

`vllm/entrypoints/pooling/base/protocol.py`

唯一变更文件，包含全部修复逻辑：新增 `padding` 字段、重构 `build_tok_params` 返回路径、追加 `pad_prompt_tokens` 封顶。

```
class EmbeddingTokenizeParamsMixin(PoolingTokenizeParamsMixin):
    # 新增字段：读取请求中的 padding 参数 (max_length / do_not_pad) ,
    # 供 build_tok_params 判断是否需要 chunked 模式做填充封顶。
    padding: Literal["max_length", "do_not_pad"] | None

    def build_tok_params(self, model_config: ModelConfig) -> TokenizeParams:
        default_max_total_tokens = model_config.max_model_len
        max_total_tokens: int | None = default_max_total_tokens
        max_output_tokens = 0

        pooler_config = model_config.pooler_config
        if pooler_config is not None:
            if pooler_config.enable_chunked_processing:
                # chunked 模式下 token 上限为 max_embed_len (例如 10000)
                max_total_tokens = pooler_config.max_embed_len
            else:
                # 非 chunked 模式：max_embed_len 作为输出 token 余量
                max_embed_len = pooler_config.max_embed_len or default_max_total_tokens
                max_output_tokens = default_max_total_tokens - max_embed_len

        # 先构造基准 TokenizeParams，后续按需覆盖填充参数
        tok_params = self._build_pooling_tok_params(
            model_config,
            add_special_tokens=self.add_special_tokens,
            max_total_tokens=max_total_tokens,
            max_output_tokens=max_output_tokens,
            max_output_tokens_param="max_model_len - max_embed_len",
        )

        # 核心修复：仅对 chunked + max_length 组合做填充封顶，
```

```

# 避免短输入按 max_embed_len 填充成大量 pad-only chunk,
# 既浪费 GPU 计算, 又会污染加权平均 embedding。
if (
    self.padding == "max_length"
    and pooler_config is not None
    and pooler_config.enable_chunked_processing
):
    max_padding_len = pooler_config.max_embed_len
    if max_padding_len is None:
        max_padding_len = default_max_total_tokens
    # 封顶为 min(max_model_len, max_embed_len):
    # 保留用户显式配置的更小 embedding 上限, 同时避免无谓超长填充。
    return tok_params.with_kwargs(
        pad_prompt_tokens=min(default_max_total_tokens, max_padding_len)
    )

return tok_params

```

评论区精华

核心讨论集中在 yewentao256 与作者 taneem-ibrahim 之间:

yewentao256: We now firstly padding, then chunk. So 513 -> 512 + 1. Shall we chunk first then padding? So 513 -> 512 + 512(1 padding to 512)

作者答复: 先 padding 再 chunk 会让 513 token 变为 512 + 1, 保持原长输入语义; 若先 chunk 再补 padding, 则需处理 1024 token 且 1-token chunk 被 pad 成 512 后会过度影响最终 embedding; 追踪每个 chunk 真实 token 数需要联动修改 padding、聚合权重、usage 报告与精度验证, 属于更广的独立改动, 窄修复保留当前顺序更安全。

yewentao256: I am not sure the 1 token request would generate correct output, could you test on this?

作者给出 H100/MRV2/bfloat16 实验: 1 token 填充结果与显式 511 pad 对照

max_abs=4.66e-10、cosine=1.0; 513 token 与 do_not_pad 对照 max_abs=9.31e-10; 与手动 512+1 加权聚合对照 max_abs=7.21e-9、cosine=1.0; 批量混合结果也与单条一致。结论是尾部 1-token chunk 被正确处理。

最终 yewentao256 给出 APPROVED: "LGTM, thanks for the work!"。

- padding 与 chunk 的执行顺序是否应交换 (design): 作者 taneem-ibrahim 答复保持当前顺序更安全: 先 chunk 再 pad 会多处理约一倍 token, 且 1-token chunk 被 pad 成 512 后可能过度影响加权 embedding; 正确实现需要联动修改 padding、真实 token 跟踪、聚合权重、usage 报告与精度验证, 属于更广的独立变更。
- 单 token 输入在 max_length 填充下是否正确 (correctness): 作者在 H100/MRV2/bfloat16 上验证: 1 token 填充输出与显式 511 pad 对照 max_abs=4.66e-10、cosine=1.0; 513 token 边界与 do_not_pad 及手动 512+1 加权聚合对照也均一致, 证明尾部单 token chunk 被正确处理。

风险与影响

• 风险：主要风险点如下：

1. 行为预期变化：对启用 `chunked_processing` 且请求 `padding="max_length"` 的用户，填充目标从 `max_embed_len` 变为 `min(max_model_len, max_embed_len)`。若某用户依赖旧行为获取固定 10000 长度输出，会收到更短序列；这是本 PR 的修复意图，但属于可感知的行为变更。
2. 数值一致性依赖人工验证：作者提供了 H100 上的对照实验（差异在 $1e-9$ 量级），但未新增自动化测试覆盖该组合路径，后续重构仍可能回归此行为，缺少测试作为护栏。
3. 范围隔离：改动集中在 `chunked + max_length` 分支，非 `chunked`、非 `max_length` 以及 `FixedMaxLenTokenizeParamsMixin` 路径均不受影响，回归面较小。
4. `max_embed_len` 为 `None` 时：`max_padding_len` 回退到 `default_max_total_tokens`，`min` 后仍为 `max_model_len`，行为与修复目标一致，无明显风险。
- 影响：影响范围限于 pooling / embedding 服务中同时开启 `chunked_processing` 与 `padding="max_length"` 的用户：短输入不再被填充到 10000 级别的聚合上限，GPU 计算量、延迟与显存占用大幅下降，usage 统计 (`prompt_tokens`) 从 10000 回归真实值 512。对系统整体吞吐有正向影响，因为不再产生整批 pad-only chunk；对非 `chunked` 或其他 `padding` 模式的请求完全无影响。团队侧改动仅为单文件 18 行，但缺少直接测试文件，需要后续维护者关注该组合路径的回归。
- 风险标记：缺少新增测试覆盖，padding 行为语义微调，数值一致性依赖人工验证

关联脉络

- PR #56760 [Bugfix] Measure complete pooling responses: 同为 pooling 前端修复，改进 pooling 服务的响应度量与基准正确性，与本 PR 同属 pooling 入口质量维护线。
- PR #57024 [Bugfix] Avoid nested score cancellation handlers for /v1/score alias: 同属 pooling/scoring API 入口的 bugfix，修复入口层装饰器行为，与本 PR 在 pooling 前端路径上相邻。