PR #38827 完整报告

vllm-project/vllm

feat: add max_tokens_per_doc in rerank request.

合并时间：2026-04-13 16:24

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/38827

执行摘要

本 PR 在 vLLM 的 rerank 和 score API 中新增 max_tokens_per_doc 和 max_tokens_per_query 参数，支持对长文档和查询进行独立截断，以对齐 Cohere、Jina 等行业标准。实现涵盖协议层扩展、三种模型类型的截断策略、全面验证和测试，作为 PR #33315 的 rebase 版本解决了目录重构导致的合并困难，并经多轮 review 优化后合并，对前端 API 和 pooling 模块有中等影响。

功能与动机

为什么做：PR body 明确指出，目的是添加 max_tokens_per_doc 和 max_tokens_per_query 参数到 rerank 和 score 请求，使 vLLM 与 Cohere、Jina 等外部 rerank API 对齐。当文档或查询过长时，能在评分前进行截断，提升处理效率和兼容性。此变更源于旧 PR #33315，但因 score/ → scoring/ 目录重构和 IO 处理器重构无法机械 rebase，故重新实现并增强。

实现拆解

实现按模块分层梳理：

1. 协议层（protocol.py）：
   - 在 ScoreRequestMixin 中添加 max_tokens_per_query 和 max_tokens_per_doc 字段，类型为 int，默认值 0（表示无截断）。
   - 使 RerankRequest 继承自 ScoreRequestMixin，移除重复的 build_tok_params 和 to_pooling_params 方法。
2. 处理层（io_processor.py）：
   - 新增 _validate_token_limit：验证参数非负且小于 max_model_len。
   - _get_token_limits：从请求或 PoolingParams 提取参数，支持在线和离线路径。
   - _truncate_scoring_data：根据限制截断文本，调用工具函数。
   - 三种截断策略：
   - 交叉编码器（带 sep token）：使用 tokenizer 的 truncation="only_second"。
   - 聊天模板 /Jinja 路径：通过 offset_mapping 在模板应用前进行文本截断，避免编码 - 解码损耗。
   - LLM-as-reranker：直接切片 token ID。

3. 工具层（utils.py）：
   - truncate_text_to_tokens：利用 offset_mapping 精确截断文本至指定 token 数。
   - get_num_special_tokens_for_pair：计算成对编码的特殊 token 数量。
4. 测试层：新增 test_max_tokens_per_doc.py 覆盖多模型场景，优化为参数化 fixture 避免 OOM。

评论区精华

review 讨论中的关键交锋：

- 参数设计：noooop 建议“将 max_tokens_per_doc 移至 ScoreRequestMixin”，以实现 /score 端点共享，并让 RerankRequest 继承以减少重复。作者采纳并回复：“Done. Moved... and RerankRequest now inherits”。
- 验证优化：noooop 指出“base/serving.py 中的验证重复”，作者移除并集中到 io_processor，确保一致性。
- 测试性能：针对测试可能 OOM 的问题，noooop 建议“使用 @pytest.fixture(scope="module", params=[.....])”，作者改为参数化运行，逐个启动服务器。
- API 限制：关于 pooling_params 是否为 list 的讨论，最终确认为“score API 不支持 Sequence[PoolingParams]”，因此添加 assert 强化设计假设。

风险与影响

技术风险：

1. 回归风险：修改核心处理逻辑可能影响现有 rerank/score 功能，但 23 个现有测试零回归提供了保障。
2. 截断精度：truncate_text_to_tokens 依赖 offset_mapping，对不支持此功能的 tokenizer 可能回退，需确保广泛兼容。
3. 验证覆盖：参数验证集中在 io_processor，需确保所有调用路径（如离线 via PoolingParams）都经过检查。

影响评估：

- 用户：提供更精细的截断控制，对齐行业实践，提升长文档处理体验。
- 系统：扩展 pooling/scoring 模块，支持多模型截断，增加约 500 行代码，复杂度可控。
- 团队：代码重构（如继承优化）提升可维护性，但新参数需文档更新和用户教育。

关联脉络

此 PR 是 PR #33315的直接后继，由于目录重构无法直接合并，故重新实现并整合了原 PR 的反馈。同时，它与 PR #39153关联，后者引入了 _params_to_seq helper，影响本 PR 中 pooling_params 处理的代码演进（最终在 rebase 后移除了临时 helper）。从近期历史 PR 看，vLLM 正在持续增强 pooling 和 scoring 功能（如 #39530 的重命名、#39655 的 bugfix），本 PR 是该趋势的一部分，旨在提升 API 兼容性和灵活性。