PR #37054 完整报告

sgl-project/sglang

Handle unlimited tokenizer context lengths

合并时间：2026-08-30 08:29

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/37054

执行摘要

- 一句话：修复 tokenize 接口对 HF 无限长标记的序列化崩溃
- 推荐动作：值得快速阅读：5 行改动展示了处理第三方库序列化约束的保守策略——只替换会出错的值而非一刀切。可关注两点：一是 ORJSON 64 位范围与 HF 无限标记的冲突是同类端点都可能踩的坑；二是建议后续补一个注册在 test/registered 规范位置的轻量单测，覆盖无限标记回退路径，避免回归。

功能与动机

PR body 明确说明：/v1/tokenize can fail with Integer exceeds 64-bit range when tokenizer.model_max_length contains Hugging Face's unlimited-length marker, as reported in #16718。设计意图是 To preserve existing behavior for non-Kimi models, this change keeps every tokenizer limit ORJSON can serialize，即只替换会破坏序列化的值，避免改变其他模型的响应语义。

实现拆解

变更入口是 python/sglang/srt/entrypoints/openai/serving_tokenize.py 中 OpenAIServingTokenize._handle_non_streaming_request。实现拆解如下：

1. 获取 max_model_len：沿用 getattr(tokenizer, "model_max_length", -1)，保持默认 -1 与既有行为一致。
2. 新增 64 位范围守卫：在构造 TokenizeResponse 之前判断，若 model_max_length 是 int 且不在 [-(263), 264) 区间内，则回退为 self.tokenizer_manager.model_config.context_len。该区间对应 ORJSON 可序列化的 signed negative / unsigned positive 64 位整数，HF 的无限标记 10**30 恰好落在区间外。
3. 行为保持策略：有限 tokenizer 限制（包括 -1、负数、2**64 以内的正数）一律原样返回，只有会触发序列化异常的值才被替换，从而不影响非 Kimi 等常规模型的响应。
4. 测试与配套：作者曾在 test/registered/unit/entrypoints/openai/test_serving_tokenize.py 新增约 83 行 CPU 测试，覆盖两条路径（有限值保留、无限值回退）；Fridge003 review 要求删除该测试，作者在提交 62eb37a 中移除，最终合入版本无测试、无配置或文档配套变更。

关键文件：

- python/sglang/srt/entrypoints/openai/serving_tokenize.py（模块 API 入口；类别 source；类型 core-logic；符号 OpenAIServingTokenize._handle_non_streaming_request）：核心修复文件，/v1/tokenize 非流式请求处理入口，新增 5 行 64 位范围守卫，是 PR 唯一合入的改动。

关键符号：OpenAIServingTokenize._handle_non_streaming_request

关键源码片段

python/sglang/srt/entrypoints/openai/serving_tokenize.py

核心修复文件，/v1/tokenize 非流式请求处理入口，新增 5 行 64 位范围守卫，是 PR 唯一合入的改动。

```
async def _handle_non_streaming_request(
    self,
    adapted_request: TokenizeRequest,
    request: TokenizeRequest,
    raw_request: Request,
) -> Union[TokenizeResponse, ErrorResponse]:
    try:
        tokenizer = self.tokenizer_manager.tokenizer
        max_model_len = getattr(tokenizer, "model_max_length", -1)

        # ORJSON 仅接受 signed 负整数与 unsigned 正整数（64 位范围）。
        # HF 的 " 无限上下文 " 标记（如 10**30）会超出该范围，
        # 直接放入响应会触发 "Integer exceeds 64-bit range" 异常。
        if isinstance(max_model_len, int) and not (
            -(2**63) <= max_model_len < 2**64
        ):
            # 仅在这种情况下回退到服务端配置的上下文长度，
            # 其余有限值（含 -1）保持原样，兼容既有调用方。
            max_model_len = self.tokenizer_manager.model_config.context_len

        if request.messages is not None:
            token_ids = self._tokenize_chat_request(request)
            tokens = token_ids
            count = len(token_ids)
        elif isinstance(request.prompt, str):
            token_ids = tokenizer.encode(
                request.prompt,
                add_special_tokens=request.add_special_tokens,
            )
            tokens = token_ids
            count = len(token_ids)
        elif isinstance(request.prompt, list):
            token_ids_list = [
                tokenizer.encode(
                    text, add_special_tokens=request.add_special_tokens
                )
```

```
                for text in request.prompt
            ]
            tokens = token_ids_list
            count = [len(ids) for ids in token_ids_list]
        else:
            return self.create_error_response(
                f"Invalid prompt type: {type(request.prompt)}. Expected str or List[str]."
            )

        return TokenizeResponse(
            tokens=tokens, count=count, max_model_len=max_model_len
        )
    except ValueError as e:
        return self.create_error_response(str(e))
```

评论区精华

唯一实质性 review 交锋发生在新增测试文件上：Fridge003 直接要求 Please remove this test，作者回复 Removed in 62eb37a20，并在 commit message 中说明是遵从维护者要求而不改变修复逻辑。该讨论的后果是 PR 最终没有任何回归测试，属于流程性决定而非技术取舍。另外，作者在 Issue 评论中请求打开 run-ci 门禁，hnyls2002 执行 /tag-and-rerun-ci 触发 CI。

- 删除新增的 serving tokenize 端点测试 (testing): 测试文件被移除，最终 PR 仅含 5 行源码改动，无任何测试配套。

风险与影响

- 风险：
  - 无测试覆盖：唯一候选测试被 Fridge003 要求删除，最终 5 行改动无任何自动化验证，后续重构 /v1/tokenize 时该边界容易回归。
  - 类型边界：回退守卫只处理 int，若 model_max_length 是 float（如 float('inf')）或其他类型，不会触发回退，ORJSON 对非有限 float 的序列化行为未在本 PR 验证。
  - 回退链路依赖：self.tokenizer_manager.model_config.context_len 需保证可用且本身在 64 位范围内，若服务端配置了超出 2**64 的异常 context_len，回退值仍可能序列化失败。
  - API 语义变化：对使用无限标记的模型，max_model_len 从报错变为返回服务端配置值，依赖旧报错行为的客户端会观察到语义变化，尽管这是修复的预期结果。
- 影响：
  - 用户面：修复了 /v1/tokenize 在部分模型（如 body 提到的 Kimi 相关 tokenizer）下的失败，响应中 max_model_len 变为合理数值。
  - 影响范围：改动集中在 OpenAIServingTokenize 的非流式处理路径，/v1/chat/completions 等其他端点不受影响。
  - 团队成本：改动小、易维护；但由于没有测试，评审者需要依赖手动验证或 CI 间接覆盖。
  - 关联问题：对应 issue #16718，是用户可见的公开 API 修复。

- 风险标记：缺少测试覆盖，公开 API 行为变更，边界类型未覆盖

关联脉络

- 暂无明显关联 PR