

PR #34488 完整报告

sgl-project/sglang

[feature] Add response-level input/output token ids to chat completions via SglExt

合并时间: 2026-09-05 07:45

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34488>

执行摘要

- 一句话: 通过 SglExt 为聊天补全添加响应级输入输出 token ID。
- 推荐动作: 这个 PR 值得精读。它解决了一个明确的、有实际需求的问题 (精确 token ID 的获取), 设计上遵循了现有 sglex 扩展的模式, 通过可选的标志和参数提供了灵活的启用方式。实现考虑周全, 特别是对流式 / 非流式、增量 / 非增量、正常 / 中止等多种场景的覆盖, 以及通过全面的单元测试保障。主要关注点在于流式路径下 ID 收集和中止处理的实现细节, 以及 split_ids 方法的设计, 这些都体现了良好的软件工程实践。

功能与动机

PR body 中明确指出, 精确的 token ID 无法通过重新标记可靠地恢复。现有的 `return_token_ids` 在聊天流式响应中不被支持 (PR#30917 添加了该标志, 但聊天流拒绝它, 且其流式形状将每个块的 ID 增量交错到每个 choice 中)。因此, 需要一个新的机制在响应级别的 `sglex` 扩展中一次性提供输入和输出 token ID。

实现拆解

1. 协议层扩展: 在 `protocol.py` 的 `SglExt` 类中添加 `input_ids` 和 `output_ids` 字段, 作为响应级别的共享数据。在 `ChatCompletionRequest` 中添加 `return_input_ids_in_sglex` 和 `return_output_ids_in_sglex` 布尔标志, 用于按请求控制是否返回。新增 `split_ids()` 方法, 将 `sglex` 对象分为包含 ID 和不包含 ID 的两部分, 为流式传输中的分离发送提供支持。
2. 服务器参数添加: 在 `server_args.py` 的 `ServerArgs` 中添加 `return_input_ids` 和 `return_output_ids` 两个布尔参数, 允许通过 `--return-input-ids` / `--return-output-ids` 在服务器启动时全局启用 ID 返回, 无需修改每个请求。
3. 服务逻辑实现: 在 `serving_chat.py` 中, 新增 `_should_return_input_ids` 和 `_should_return_output_ids` 两个辅助方法, 统一判断是否应返回 ID (综合请求标志和服务端参数)。在 `_convert_to_internal_request` 中, 支持通过 HTTP 头 `x-sglex-return-input-ids` 和 `x-sglex-return-output-ids` 进行按请求启用。在 `_generate_chat_stream` 流式路径中, 新增状态变量 `input_ids`、`output_ids` 和 `error_aborted`, 用于在流式迭代中收集 ID 并处理中止情况。在 `_build_chat_response` 非流式路径中, 直接将收集到的 ID 设置到响应的 `sglex` 字段中。
4. 中止处理修复: 修改 `tokenizer_manager.py` 中的 `_handle_abort_req` 方法, 确保在中止时, `prompt_token_ids` 被包含在中止响应中 (当请求要求时), 并且 `output_ids` 的截断逻辑仅适用于增量流式输出模式, 避免非增量模式下的数据丢失。

5. 测试覆盖：在 `test_serving_chat.py` 中添加了全面的单元测试，覆盖了非流式下 ID 的返回与默认不返回、流式下 ID 的收集、增量与非增量模式下 ID 的处理、以及中止场景下 ID 不被泄露到错误块等关键路径。在 `test_tokenizer_manager_rid_cleanup.py` 中新增了 `TestAbortOutputPayload` 测试类，验证中止块是否正确包含 `prompt_token_ids` 以及 `output_ids` 的行为是否与流式模式匹配。

关键文件：

- `python/sglang/srt/entrypoints/openai/serving_chat.py`（模块 聊天服务；类别 source；类型 core-logic；符号 `_should_return_input_ids`, `_should_return_output_ids`）：核心变更文件，实现了在流式和非流式路径中收集、处理和发送输入输出 token ID 的主要逻辑。
- `python/sglang/srt/entrypoints/openai/protocol.py`（模块 协议定义；类别 source；类型 data-contract；符号 `split_ids`）：协议定义文件，新增了 `sglxt` 的 ID 字段和 `ChatCompletionRequest` 的请求标志，定义了数据契约。
- `test/registered/unit/entrypoints/openai/test_serving_chat.py`（模块 聊天服务测试；类别 test；类型 test-coverage；符号 `test_non_streaming_ids_emit_sglxt`, `test_streaming_abort_with_ids_enabled`, `test_streaming_error_abort_still_finalizes_other_choices`）：核心测试文件，提供了对新功能在非流式、流式、增量、非增量、正常、中止等多种场景下的单元测试覆盖。
- `python/sglang/srt/managers/tokenizer_manager.py`（模块 Tokenizer 管理器；类别 source；类型 core-logic）：修改了中止处理逻辑，确保 `prompt_token_ids` 和正确的 `output_ids` 在中止响应中被携带。
- `python/sglang/srt/server_args.py`（模块 服务参数；类别 config；类型 configuration）：添加了服务器全局参数 `--return-input-ids` 和 `--return-output-ids`，提供配置灵活性。

关键符号：`_should_return_input_ids`, `_should_return_output_ids`, `split_ids`, `_handle_abort_req`

关键源码片段

`python/sglang/srt/entrypoints/openai/serving_chat.py`

核心变更文件，实现了在流式和非流式路径中收集、处理和发送输入输出 token ID 的主要逻辑。

```
# 从 serving_chat.py 整理的核心辅助方法，用于判断是否应返回 token ID。
def _should_return_input_ids(self, request: ChatCompletionRequest) -> bool:
    """判断是否应在响应级 sglxt 中返回提示（输入） token ID。"""
    # 综合判断：请求级标志 OR 服务器全局参数
    return (
        request.return_input_ids_in_sglxt
        or self.tokenizer_manager.server_args.return_input_ids
    )

def _should_return_output_ids(self, request: ChatCompletionRequest) -> bool:
    """判断是否应在响应级 sglxt 中返回采样输出 token ID。"""
    # 综合判断：请求级标志 OR 服务器全局参数
    return (
        request.return_output_ids_in_sglxt
```

```
        or self.tokenizer_manager.server_args.return_output_ids
    )
```

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

协议定义文件，新增了 `sgllex` 的 ID 字段和 `ChatCompletionRequest` 的请求标志，定义了数据契约。

从 protocol.py 整理的 SglExt 类扩展及 split_ids 方法。

```
class SglExt(BaseModel):
```

```
    """SGLang 扩展字段，用于 OpenAI 兼容的响应。"""
```

```
    # ... 现有字段 ...
```

```
    input_ids: Optional[List[int]] = None # 共享的提示 token ID
```

```
    output_ids: Optional[List[List[int]]] = None # 每个 choice 的输出 token ID
```

```
def split_ids(self) -> Tuple[Optional["SglExt"], Optional["SglExt"]]:
```

```
    """将设置的字段拆分为 (非 id 部分, id 部分);
```

```
    某一侧若无设置字段则返回 None。"""
```

```
    non_ids: Dict[str, Any] = {}
```

```
    ids: Dict[str, Any] = {}
```

```
    for name in type(self).model_fields:
```

```
        value = getattr(self, name)
```

```
        if value is None:
```

```
            continue
```

```
        # 将 input_ids 和 output_ids 分到 ids 组，其余分到 non_ids 组
```

```
        (ids if name in ("input_ids", "output_ids") else non_ids)[name] = value
```

```
    return (
```

```
        type(self)(**non_ids) if non_ids else None,
```

```
        type(self)(**ids) if ids else None,
```

```
)
```

test/registered/unit/entrypoints/openai/test_serving_chat.py

核心测试文件，提供了对新功能在非流式、流式、增量、非增量、正常、中止等多种场景下的单元测试覆盖。

从 test_serving_chat.py 整理的一个代表性测试用例，

验证在启用了 ID 返回时，流式错误中止后不应泄露 ID 数据。

```
def test_streaming_abort_with_ids_enabled(self):
```

```
    """测试在启用 ID 时，一个终端错误中止只产生错误块和 [DONE]。"""
```

```
    err_msg = "Aborted by scheduler"
```

```
    err_code = HTTPStatus.INTERNAL_SERVER_ERROR
```

模拟一个立即以错误中止的生成器

```
async def _mock_generate_abort():
```

```
    yield {
```

```
        "text": "Partial ",
```

```
        "prompt_token_ids": [4, 5, 6], # 包含提示 ID
```

```
        "output_ids": [1, 2, 3], # 包含输出 ID
```

```
        "meta_info": { ... "finish_reason": {"type": "abort", "status_code": err_code, ...} },
```

```
        "index": 0,
```

```

    }
    # ... 设置请求和 mock ...

    # 执行流式生成
    loop = get_or_create_event_loop()
    chunks = loop.run_until_complete(run_stream())

    # 验证: 只有错误块和 [DONE], 没有包含 input_ids 或 output_ids 的块
    self.assertIn("error", chunks[0])
    self.assertEqual(chunks[1], "data: [DONE]")

")
    self.assertFalse(
        any("input_ids" in c or "output_ids" in c for c in chunks),
        "sglext ids event leaked after abort error",
    )
)

```

评论区精华

1. 字段命名争议: @guapisolo 指出新标志可能与现有的 `return_prompt_token_ids` / `return_token_ids` 语义重叠, 建议重命名为 `return_input_ids_in_sglext` / `return_output_ids_in_sglext` 以明确其作用于响应级 `sglext` 扩展。作者 @amykchang 采纳了该建议并完成了重命名。
 2. 非增量模式下的中止处理: @JustinTong0323 指出在默认的非增量模式下, 中止块可能被截断为 `[last_token]`, 导致之前生成的精确 token ID 丢失。作者修复了 `tokenizer_manager.py` 和 `serving_chat.py` 中的逻辑, 确保非增量中止块携带完整的累积 `state.output_ids`。
 3. 错误中止对流终结的影响: @JustinTong0323 指出最初实现中错误中止后直接返回会丢弃其他 `choice` 的终结块和使用统计, 可能破坏现有流式协议。作者恢复了 `break` 行为, 并添加了 `error_aborted` 标志, 仅在出错时不发送 `sglext` 中的 token ID, 保留了原有的流终结流程。
 4. 缓冲区压力下的合并机制: @dcw02 提醒注意 `tokenizer_manager.py` 中的 `_coalesce_streaming_chunks` 机制, 该机制在背压下会合并排队的块。最终实现确保 ID 的收集逻辑与此机制兼容, 尤其是在中止场景下。
- 新请求标志的命名与语义 (design): 作者采纳建议, 完成重命名。这明确了新标志的语义, 避免了 API 消费者混淆。
 - 非增量模式下中止块的 `output_ids` 完整性 (correctness): 作者修复了 `tokenizer_manager._handle_abort_req`, 仅在增量流式模式下截断 `output_ids`。非增量模式现在返回完整的累积 `state.output_ids`。测试 `test_abort_output_ids_match_the_streaming_mode` 验证了此行为。
 - 错误中止对流终结协议的影响 (correctness): 作者恢复了 `break` 行为, 添加了 `error_aborted` 标志, 仅在出错时跳过 `sglext` 中的 token ID 发送, 保留了原有流终结流程。测试 `test_streaming_error_abort_still_finalizes_other_choices` 覆盖了此场景。

- 背压合并机制下的 `prompt_token_ids` 丢失 (correctness): 最终实现直接在 `_handle_abort_req` 中附加 `state.prompt_token_ids` 到中止输出, 避免了依赖可能被合并的流块。这保证了在所有中止路径下 `sglex.input_ids` 都能正确返回。

风险与影响

- 风险:
 1. 核心路径变更: `serving_chat.py` 中的 `_generate_chat_stream` 和 `_build_chat_response` 是聊天补全的核心响应路径, 任何错误都可能导致响应失败或数据损坏。本次变更在这两个方法中增加了条件逻辑和状态变量, 需要仔细验证。
 2. 中止边界处理: 中止场景下的行为是本 PR 修复的重点, 但也增加了复杂性。需要确保在所有中止类型 (如错误中止、优雅中止) 和流式模式 (增量 / 非增量) 下, ID 数据都保持一致且正确。修复后的逻辑依赖于 `incremental_streaming_output` 配置, 需确认该配置在所有部署环境中的一致性。
 3. 流式语义变更: 虽然 `break` 行为被恢复, 但错误中止后不再发送 `sglex` 中的 token ID 是一个语义变更。需要评估依赖此信息的现有客户端或监控工具是否受影响。
 4. 向后兼容性: 新增的协议字段和请求头是可选且向后兼容的。默认情况下不返回 ID, 不会影响现有客户端。服务器全局参数 `--return-input-ids / --return-output-ids` 默认关闭, 但启用后会改变所有请求的响应内容。
- 影响:
 1. 用户影响: 为需要精确 token 级别信息的下游任务 (如 token 级别对齐、调试、分析) 提供了可靠的数据源。用户现在可以通过请求头或服务器参数灵活启用此功能, 覆盖流式和非流式场景。
 2. 系统影响: 在 OpenAI 兼容的 API 层添加了新的可选字段和处理逻辑。对系统性能的影响微乎其微, 因为 ID 信息本身是现有生成过程的副产品, 收集和序列化开销很小。
 3. 团队影响: 增加了 API 的可观察性和调试能力。测试覆盖的增加提升了聊天补全核心路径的可靠性。需要文档化新字段和参数的使用方式。 - 风险标记: 核心路径变更, 中止边界处理, 流式语义变更

关联脉络

- PR #30917 Add `return_token_ids` for chat completion: 本 PR 直接提及 PR#30917。该 PR 添加了 `return_token_ids` 标志, 但其在聊天流式中的实现有缺陷 (每个 choice 交错 ID 增量), 本 PR 是对其的替代和完善, 提供了更清晰、更可靠的响应级 token ID 返回机制。