PR #23751 完整报告

sgl-project/sglang

[3/N][Sync sglang-miles] TITO Support

合并时间：2026-06-04 09:45

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23751

执行摘要

- 一句话：新增 TITO 风格 chat：支持 pre-tokenized input_ids 和返回 prompt_token_ids
- 推荐动作：建议精读 serving_chat.py 中 _convert_to_internal_request 和 _build_chat_response 的实现，了解请求转换管线中字段优先级和错误处理的权衡。protocol.py 的自定义序列化方式也值得参考。对于部署运维，需注意 skip_tokenizer_init 场景的兼容性。

功能与动机

PR body 说明 Synced from https://github.com/sgl-project/sglang/commit/69018ba16e5d8ef56559d076066377c127fedb27 和 https://github.com/sgl-project/sglang/commit/3606aecea3215ece32fb774f0b8e2a7c780df741，目标是支持 TITO 风格的 chat 请求，具体包括：

1) 支持通过 input_ids 传入预分词 token，跳过 chat 模板； 2) 允许非流式 chat 响应返回 prompt_token_ids 和可选的 meta_info。

实现拆解

1. 协议层扩展（python/sglang/srt/entrypoints/openai/protocol.py）：在 ChatCompletionRequest 中新增 input_ids（Optional[List[int]]）、return_prompt_token_ids（bool）和 return_meta_info（bool）字段；在 ChatCompletionResponseChoice 中新增 prompt_token_ids（Optional[List[int]]）和 meta_info（Optional[Dict[str, Any]]）字段，并通过自定义 model_serializer 实现 None 字段自动排除，保持向后兼容。
2. 请求转换与拒绝规则（python/sglang/srt/entrypoints/openai/serving_chat.py）：在 _convert_to_internal_request 中，首先检查 stream=True 时是否同时设置了 return_prompt_token_ids 或 return_meta_info，若是则抛出 ValueError；在 _process_messages 中，当 request.input_ids 不为 None 时跳过 chat 模板，直接以 input_ids 作为 prompt_ids；在 prompt_kwargs 构建中优先使用 request.input_ids 路径（置于 is_multimodal 判断之前），确保多模态模型也能正确处理预分词输入；在 _build_chat_response 中根据 request 标记从 ret_item 提取对应字段并填入 choice 对象。
3. 状态管理（python/sglang/srt/managers/tokenizer_manager.py）：在 ReqState 中新增 prompt_token_ids 字段用于暂存 tokenize 后的 prompt token IDs；在

generate_request 和 _handle_batch_request 中，若 obj.return_prompt_token_ids 为 True 则将 tokenized_obj.input_ids 的副本存入 state；在 _handle_batch_output 中当 out_dict 非空且 state 有值时将 prompt_token_ids 加入输出字典。

4. 内部数据结构（python/sglang/srt/managers/io_struct.py）：在 GenerateReqInput 和 EmbeddingReqInput 中新增 return_prompt_token_ids 字段，并在各自的 __getitem__ 方法中同步传递，确保 n>1 或批处理场景下该标志不会丢失。
5. 配套测试：test_protocol.py 验证协议字段序列化；test_serving_chat.py 验证流式拒绝、input_ids 跳过模板、非流式响应的 prompt_token_ids 返回；test_io_struct.py 验证 __getitem__正确保留标志。

关键文件：

- python/sglang/srt/entrypoints/openai/serving_chat.py（模块 请求处理；类别 source；类型 core-logic；符号 _convert_to_internal_request, _process_messages, _build_chat_response）：核心变更文件，实现 TITO 请求转换、流式拒绝、input_ids 优先级和多模态兼容逻辑。
- python/sglang/srt/entrypoints/openai/protocol.py（模块 协议层；类别 source；类型 core-logic；符号 ChatCompletionRequest, ChatCompletionResponseChoice）：协议层定义新增字段和序列化逻辑，是 TITO 的数据契约基础。
- python/sglang/srt/managers/tokenizer_manager.py（模块 分词管理；类别 source；类型 core-logic；符号 ReqState, generate_request, _handle_batch_request, _handle_batch_output）：在 tokenize 后捕获 prompt token IDs 并存入状态，输出时附加到响应。
- python/sglang/srt/managers/io_struct.py（模块 IO 结构；类别 source；类型 core-logic；符号 GenerateReqInput, EmbeddingReqInput）：在 GenerateReqInput 和 EmbeddingReqInput 中新增 return_prompt_token_ids 字段并确保 __getitem__传递。
- test/registered/unit/entrypoints/openai/test_serving_chat.py（模块 测试；类别 test；类型 test-coverage；符号 test_convert_to_internal_request_rejects_stream_return_prompt_token_ids, test_convert_to_internal_request_rejects_stream_return_meta_info, test_convert_to_internal_request_input_ids_bypasses_template, test_non_streaming_chat_response_returns_requested_prompt_ids_and_meta_info）：测试流式拒绝、input_ids 跳过模板、非流式响应返回 prompt_token_ids 和 meta_info。
- test/registered/unit/entrypoints/openai/test_protocol.py（模块 测试；类别 test；类型 test-coverage；符号 test_chat_completion_tito_extensions, test_prompt_token_ids_and_meta_info_serialization）：测试协议新字段的序列化和反序列化行为。
- test/registered/unit/managers/test_io_struct.py（模块 测试；类别 test；类型 test-coverage；符号 test_getitem_preserves_return_prompt_token_ids）：验证 GenerateReqInput.__getitem__正确传递 return_prompt_token_ids。

关键符号：_convert_to_internal_request, _process_messages, _build_chat_response, generate_request, _handle_batch_request, _handle_batch_output, test_convert_to_internal_request_rejects_stream_return_prompt_token_ids, test_chat_completion_tito_extensions, test_prompt_token_ids_and_meta_info_serializatio

n, test_getitem_preserves_return_prompt_token_ids

关键源码片段

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

核心变更文件，实现 TITO 请求转换、流式拒绝、input_ids 优先级和多模态兼容逻辑。

```
# _convert_to_internal_request 方法中的流式限制和 input_ids 优先处理
if request.stream:
    if request.return_prompt_token_ids:
        raise ValueError(
            "return_prompt_token_ids is not supported with streaming. "
            "Please set stream=false when using return_prompt_token_ids=true."
        )
    if request.return_meta_info:
        raise ValueError(
            "return_meta_info is not supported with streaming. "
            "Please set stream=false when using return_meta_info=true."
        )

is_multimodal = self.tokenizer_manager.model_config.is_multimodal
processed_messages = self._process_messages(request, is_multimodal)
sampling_params = request.to_sampling_params(
    stop=processed_messages.stop,
    model_generation_config=self.default_sampling_params,
    tool_call_constraint=processed_messages.tool_call_constraint,
)

# 优先使用用户提供的 input_ids，避免多模态路径覆盖
if request.input_ids is not None:
    prompt_kwargs = {"input_ids": processed_messages.prompt_ids}
elif is_multimodal:
    prompt_kwargs = {"text": processed_messages.prompt}
else:
    if isinstance(processed_messages.prompt_ids, str):
        prompt_kwargs = {"text": processed_messages.prompt_ids}
    else:
        prompt_kwargs = {"input_ids": processed_messages.prompt_ids}
```

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

协议层定义新增字段和序列化逻辑，是 TITO 的数据契约基础。

```
class ChatCompletionResponseChoice(BaseModel):
    # ... 其他字段 ...
    # 新增可选字段，仅当请求要求时才填充
    prompt_token_ids: Optional[List[int]] = None
    meta_info: Optional[Dict[str, Any]] = None

    @model_serializer(mode="wrap")
```

```
def _serialize(self, handler):
    data = handler(self)
    # 排除 None 字段，保持 response 干净且向后兼容
    if self.hidden_states is None:
        data.pop("hidden_states", None)
    if self.prompt_token_ids is None:
        data.pop("prompt_token_ids", None)
    if self.meta_info is None:
        data.pop("meta_info", None)
    return data
```

评论区精华

1. content 默认值变更：guapisolo 提出将 content 从 None 改为 "" 是否可接受，担心 chat模板将None渲染为"None"。JustinTong0323表示赞同（+1）。最终该变更被接受。
2. multimodal 路径优先级：JustinTong0323 指出在 is_multimodal=True 时 input_ids 路径会被 text="" 覆盖，导致模型收到空 prompt。作者随后调整了条件判断顺序，优先使用 request.input_ids。
3. batch 子请求字段传递：chatgpt-codex 报告 GenerateReqInput.__getitem__ 未复制 return_prompt_token_ids，导致批量请求中该标志恒为 False。JustinTong0323 建议向 EmbeddingReqInput 也添加该字段以避免使用 getattr。最终通过显式添加字段解决。
4. skip_tokenizer_init 兼容性：chatgpt-codex 指出 input_ids 分支中调用了 tokenizer.decode，在 skip_tokenizer_init=True 部署下 tokenizer 可能为 None，导致崩溃。作者需要额外处理此情况（当前 patch 未体现，可能已修改或待后续）。

- Content 默认值变更：None → "" (design): 接受变更，认为改为空串更合理。
- Multimodal 路径优先级 (correctness): 作者将 input_ids 检查提前到 is_multimodal 之前，确保预分词输入优先。
- Batch 子请求 missing return_prompt_token_ids (correctness): 显式添加字段到 GenerateReqInput 和 EmbeddingReqInput 的 getitem。
- skip_tokenizer_init 兼容性 (correctness): 未直接在当前 patch 中修复，需后续关注。
- 使用 getattr 还是显式字段 (style): 采用显式字段方案。

风险与影响

- 风险：
  - 批量请求数据一致性：若 GenerateReqInput.__getitem__ 未正确传递 return_prompt_token_ids，则 n>1 的请求会静默丢失 prompt token IDs。已通过显式添加字段和测试缓解。
  - multimodal 模型空 prompt：若 input_ids 路径在 is_multimodal 检查之前执行则正常，但原始代码中 multimodal 分支后于 input_ids 分支，可能存在顺序问题，已修复。
  - skip_tokenizer_init 部署失败：如 review 所述，当 tokenizer 为 None 时调用 .decode() 会崩溃。此风险需要确认当前实现是否处理（patch 中可能未处理，建议关注）。

    - content 默认值变更：将 None 改为 "" 可能影响依赖 None 检查的客户端或下游处理，但 chat 模板渲染行为得到改善（避免 "None" 字符串）。
- 影响：
    - 用户影响：新功能对现有请求完全向后兼容，新增字段默认均为 False 或 None。需要 TITO 能力的用户可以开始使用 input_ids 和 return_prompt_token_ids 参数。
    - 系统影响：变更集中在请求处理管线，不涉及模型推理、KV Cache 等性能敏感路径，对吞吐和延迟无显著影响。增加了一个存储 prompt token IDs 的字段和条件拷贝，内存开销极小。
    - 团队影响：需要维护新增字段的一致性（尤其是 __getitem__ 和序列化逻辑），增加少量技术债务。
    - 风险标记：batch 数据一致性，multimodal 路径矛盾，tokenizer 依赖崩溃，content 默认值变更

关联脉络

- 暂无明显关联 PR