PR #44683 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Fix missing added tokens in hf/fastokens tokenizer

合并时间：2026-06-10 18:52

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44683

PR 分析报告：修复 Rust 前端 Tokenizer 遗漏 tokenizer_config.json 中的 added tokens

执行摘要

此 PR 修复了 Rust 前端加载 HuggingFace 模型时，tokenizer 只读取 tokenizer.json 而遗漏 tokenizer_config.json 中 added_tokens_decoder 定义的特殊 token（如 <|image_pad|>）的关键 bug。通过新增合并模块，在加载 tokenizer 前将两个配置文件中的 token 整合，使得 Qwen2-VL 等多模态模型能在 Rust 前端正工作。变更集中且影响面有限（仅 Rust 前端），设计稳健，并补充了相应测试。

功能与动机

对于 Qwen/Qwen2-VL-2B-Instruct 等多模态模型，其图像占位符 token <|image_pad|> 仅定义在 tokenizer_config.json 的 added_tokens_decoder 中，而不在 tokenizer.json 中。Rust 前端（fastokens 或 HuggingFace tokenizers）加载 tokenizer 时仅读取 tokenizer.json，导致该 token 缺失，进而引发 multimodal 预处理错误：

> placeholder token <|image_pad|> is not in the tokenizer vocabulary

此 PR 解决了该问题，使 Rust 前端能够正确加载所有 added tokens。

实现拆解

1. 新增 rust/src/tokenizer/src/hf/added_tokens.rs 模块：
   - 定义 TokenizerJson、TokenizerConfigJson、AddedTokenConfig 结构体，分别投影 tokenizer.json 和 tokenizer_config.json 的 added tokens 部分，并使用 #[serde(flatten)] 保留未建模字段。
   - 提供 load_tokenizer_json_with_extra_tokens 函数：读取 tokenizer.json 后，查找同目录下的 tokenizer_config.json，若存在则解析并调用 merge_added_tokens_from_config 合并。
   - 合并逻辑：遍历 added_tokens_decoder 的键值对，将每个键解析为 u32 作为 token id，若该 id 尚未出现在 tokenizer.json 的 added_tokens 中，则创建一个带有 id 的 AddedTokenConfig 并追加。
2. 修改 rust/src/tokenizer/src/hf.rs 中的构造方法：

- 将 new_fastokens 和 new_hf 从直接通过路径加载 tokenizer（FastokensTokenizer::from_file / HfTokenizer::from_file）改为先调用 load_tokenizer_json_with_extra_tokens 获取合并后的 JSON，再通过 from_json / from_value 构建底层 tokenizer。
- 这样 fastokens 和 HuggingFace 两种后端都能获得完整的 added tokens。

3. 修改 rust/src/chat/src/multimodal.rs 的可见性：
   - 将 placeholder_token 方法从 pub(crate) 改为 pub，以便测试代码可以通过 MultimodalModelInfo 获取占位符 token。
4. 更新测试文件 rust/src/server/src/routes/tests.rs：
   - 在 FakeChatTokenizer::encode 中添加对 <|image_pad|> 的分支处理（token id 151655）。
   - 在 ChatRenderer::render 中，使用从 MultimodalModelInfo::placeholder_token() 获取的占位符 token 替换硬编码的 <image>。
   - 修正 qwen_multimodal_model_info 测试 fixture 中的字段名从 vision_token_id 改为 image_token_id。
5. 测试配套：
   - added_tokens.rs 包含单元测试 merge_added_tokens_from_config_preserves_unmodeled_fields，验证合并后未建模字段仍被保留。
   - hf.rs 包含集成测试 constructors_merge_extra_added_tokens_from_tokenizer_config，验证 fastokens 和 hf 两种后端均能正确识别合并后的 token <|image_pad|>。

hf.rs — 构造方法修改（示例）

```
use crate::hf::added_tokens::load_tokenizer_json_with_extra_tokens;

pub fn new_fastokens(path: &Path) -> Result<Self> {
    info!(path = %path.display(), "loading tokenizer with fastokens");
    let tokenizer_json = load_tokenizer_json_with_extra_tokens(path)?;
    let t = FastokensTokenizer::from_json(tokenizer_json)
        .map_err(|error| tokenizer_error!("failed to load tokenizer: {}", error.as_report()))?;
    Ok(Self::from_fastokens_backend(t))
}

pub fn new_hf(path: &Path) -> Result<Self> {
    info!(path = %path.display(), "loading tokenizer with huggingface tokenizers");
    let tokenizer_json = load_tokenizer_json_with_extra_tokens(path)?;
    let t = serde_json::from_value::<HfTokenizer>(tokenizer_json)
        .map_err(|error| tokenizer_error!("failed to load tokenizer: {}", error.as_report()))?;
    Ok(Self::from_hf_backend(t))
}
```

更多完整实现请参阅 key_files 中的 added_tokens.rs 代码块。

评论区精华

> Isotr0py: "BTW, this PR needs another fix from llm-multimodal side: https://github.com/vllm-project/llm-multimodal/pull/1" BugenZhao: "Good catch 🤔 Shall we also open a PR on fastokens upstream?" Isotr0py: "Sure! Opened https://github.com/crusoecloud/fastokens/pull/36 to fix this at fastokens side as well" BugenZhao: "I just realized this is not an issue specific to fastokens: HF tokenizers also only checks the content of tokenizer.json without tokenizer_config.json, while Python transformers' from_pretrained does handle that. The closest Rust implementation I can find that handles this is Dynamo."

这些讨论表明该问题不仅影响 fastokens，也影响 HuggingFace tokenizers，本 PR 从 Rust 端统一解决了两个后端。

风险与影响

风险：

- 若 tokenizer_config.json 格式异常或缺失，代码会回退并输出警告，不会崩溃，属于安全降级。
- 合并时基于 id 去重，若两个配置文件中有相同 id 但不同内容，后出现的会被忽略，这可能与用户预期不一致。
- 对非数字 key 的解析静默跳过，缺乏日志。

影响：

- 范围：仅 Rust 前端（vLLM v1）用户，且仅多模态模型依赖 tokenizer_config.json 定义特殊 token 时受影响。
- 程度：修复前这些模型无法启动，修复后可正常推理，影响重大但局限。
- 性能：增加一次文件读取与 JSON 解析，但加载仅一次，开销可忽略。

关联脉络

此 PR 与 fastokens 上游 PR #36 以及 llm-multimodal 仓库的配套修复构成问题修复链条。在 vLLM 仓库内，此前已有 #45073 等模型 bugfix，但本 PR 是首次触及 Rust 端 tokenizer 加载机制，为后续多模态模型在 Rust 前端的支持奠定了基础。