PR #40582 完整报告

vllm-project/vllm

Fix Cohere ASR after HF upgrade

合并时间：2026-04-30 14:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40582

执行摘要

- 一句话：修复 Cohere ASR 因 HF 升级导致的 token 编码问题
- 推荐动作：值得精读，特别是 get_generation_prompt 的重构以及如何绕过 Fast tokenizer 的限制。对维护多模态和 ASR 模型的开发者有参考价值。

功能与动机

PR 指出 Tokenization was incorrectly working on v0.19.1 due to the HF upgrade。HF 团队为模型使用的 SentencePiece tokenizer 添加了 Fast tokenizer 支持，因此需要切换到 Fast 实现。同时希望重新启用 CohereASR 的 e2e 测试，因为这是唯一一个支持变长编码器输入的 encoder-decoder ASR 模型。

实现拆解

1. 重构 prompt 生成逻辑：在 cohere_asr.py 的 CohereAsrForConditionalGeneration 类中，get_generation_prompt 方法被重写。原来使用字符串拼接的 TextPrompt 被替换为 TokensPrompt，并新增 _get_default_prompt_tokens 和 _get_default_prompt_token_ids 类方法来生成 token 级别的控制标签序列。这是因为 Fast SentencePiece tokenizer 对以特定前缀（如 __）开头的字符串编码不可靠，而 token 级别构造可以精确控制编码结果。
2. 添加 token ID 缓存：引入 _default_prompt_token_ids_cache 类变量缓存已生成的 token ID，避免每次推理重复计算。
3. 修复 is_mm_prefix_lm 方法：在 model_arch_config_convertor.py 中为 CohereAsrModelArchConfigConvertor 添加 is_mm_prefix_lm 方法返回 False，因为 CohereASR 不是 prefix LM。该方法之前已被重构（PR #40701），本 PR 补齐了缺失的覆盖。
4. 启用 e2e 测试并调整测试：在 tests/models/registry.py 中移除 is_available_online=False 标志，在 test_transcription_api_correctness.py 中取消注释 CohereASR 测试用例。同时，由于测试中的 dither 随机性导致 test_processing_correctness 不稳定，添加了跳过该模型的逻辑，并改进断言消息以包含更多参数便于调试。

关键文件：

- vllm/model_executor/models/cohere_asr.py（模块 模型实现；类别 source；类型 data-contract；符号 get_generation_prompt, _get_default_prompt_tokens, _get_default_prompt_token_ids）：核心变更文件，重写了 prompt 生成逻辑，解决了 tokenization 问题并启用了 Fast tokenizer。
- tests/models/multimodal/processing/test_common.py（模块 测试公共；类别 test；类型 test-coverage）：测试调整，跳过 CohereASR 的 processing_correctness 测试并改进断言信息，
- vllm/transformers_utils/model_arch_config_convertor.py（模块 配置转换；类别 source；类型 data-contract；符号 is_mm_prefix_lm）：为 CohereASR 添加缺失的 is_mm_prefix_lm 方法，避免上游重构导致的错误。
- tests/models/registry.py（模块 测试注册；类别 test；类型 test-coverage）：移除 is_available_online=False，使 CohereASR 可用于在线推理测试。
- tests/entrypoints/openai/correctness/test_transcription_api_correctness.py（模块 正确性测试；类别 test；类型 test-coverage）：重新启用 CohereASR 的正确性测试用例。

关键符号：get_generation_prompt, _get_default_prompt_tokens, _get_default_prompt_token_ids, is_mm_prefix_lm

关键源码片段

vllm/model_executor/models/cohere_asr.py

核心变更文件，重写了 prompt 生成逻辑，解决了 tokenization 问题并启用了 Fast tokenizer。

```
# CohereAsrForConditionalGeneration 类中重构的 prompt 生成方法

@classmethod
def get_generation_prompt(cls, stt_params: SpeechToTextParams) -> PromptType:
    audio = stt_params.audio
    stt_config = stt_params.stt_config
    language = stt_params.language
    model_config = stt_params.model_config

    if language is None:
        raise ValueError("Language must be specified when creating the CohereASR prompt")

    # 获取 tokenizer 实例，用于 token ID 编码
    tokenizer = cached_tokenizer_from_config(model_config)

    # prompt_text 置为 None，因为 CohereASR 使用 fast SentencePiece tokenizer，
    # 其对第一个字符 "▁" 的处理与预期不一致，故采用 token ID 方式构造 prompt
    prompt_text = None
    # 调用类方法生成默认的 prompt token IDs
    prompt_token_ids = cls._get_default_prompt_token_ids(
        tokenizer,
        model_config,
        language,
    )
```

```
    return TokensPrompt(
        prompt=prompt_text,
        prompt_token_ids=prompt_token_ids,
        multi_modal_data={"audio": (audio, stt_config.sample_rate)},
    )


@classmethod
def _get_default_prompt_tokens(cls, language: str) -> tuple[str, ...]:
    """构造 token 级别的控制标签序列，避免 fast tokenizer 解析原始字符串时的前缀丢失问题。"""
    # 语言标签，例如 "<|en|><|en|>"
    language_tag = f"<|{language}|><|{language}|>"
    # 标点与符号控制（目前固定为 True）
    pnc = True
    pnc_tag = "<|pnc|>" if pnc else "<|nopnc|>"
    # 构建完整 token 序列（均为特殊 token，不会被 fast tokenizer 特殊处理）
    tokens = (
        "<|startofcontext|>",
        "<|startoftranscript|>",
        "<|emo:undefined|>",
        language_tag,
        pnc_tag,
        "<|noitn|>",
        "<|notimestamp|>",
        "<|nodiarize|>",
    )
    return tokens


@classmethod
def _get_default_prompt_token_ids(
    cls,
    tokenizer,
    model_config,
    language: str,
) -> tuple[int, ...]:
    # 检查缓存中是否已有该语言对应的 token IDs
    cache_key = (language,)
    if cache_key in cls._default_prompt_token_ids_cache:
        return cls._default_prompt_token_ids_cache[cache_key]

    # 获取 token 序列
    tokens = cls._get_default_prompt_tokens(language)
    # 对第一个 token 单独 encode，避免 lossy 前缀剥离
    first_id = tokenizer.encode(tokens[0], add_special_tokens=False)[0]
    # 对其余 token 整体 encode
    rest_ids = tokenizer.encode(
        tokens[1:], add_special_tokens=False, is_split_into_words=True
    )
```

```
    # 合并为完整的 IDs 序列
    token_ids = (first_id,) + tuple(rest_ids)
    # 写入缓存
    cls._default_prompt_token_ids_cache[cache_key] = token_ids
    return token_ids
```

注意：上述 _get_default_prompt_tokens 返回的 token 序列实际在最终代码中可能略有不同（如 pnc_tag 条件），这里展示核心设计思想。

评论区精华

- 自定义 prompt 的 token stripping 问题：gemini-code-assist 指出自定义 prompt 同样会遭遇快速 tokenizer 剥离前缀 __ 的问题，认为用户无法提供完全有效的自定义 prompt。作者 ekagra-ranjan 回应“removed custom prompt”，确认已移除自定义 prompt 支持，避免该问题。
- 实现参考：作者在评论中说明 _get_default_prompt_tokens 的逻辑参考了 transformers 源码中 CohereASRProcessor 的实现，贴出了具体链接。
  - 自定义 prompt 的 tokenizer stripping 问题 (design): 作者移除了自定义 prompt 支持，避免该问题。

风险与影响

- 风险：
  - Tokenizer 行为兼容性：新 tokenizer 行为依赖于 Fast SentencePiece 的具体实现，未来 transformers 升级可能再次影响。
  - 自定义 prompt 缺失：移除了自定义 prompt 支持，可能限制用户的高级用法，但当前在线推理不需要自定义 prompt。
  - 测试不稳定：test_processing_correctness 中对 CohereASR 跳过了，该模型的测试覆盖率不完整，可能遗漏回归。
- 影响：
  - 用户：Cohere ASR 模型现在可在在线推理中正常使用，不再因 tokenization 错误失败。
  - 系统：没有性能或安全影响，引入了一个类变量缓存，内存占用可忽略。
  - 团队：提供了为 encoder-decoder ASR 模型处理 tokenization 的参考实现，便于后续维护。
  - 风险标记：升级兼容性风险，tokenizer 行为依赖，测试覆盖不完全

关联脉络

- PR #40701 [Refactor] is_mm_prefix_lm refactored: 本 PR 为 CohereASR 添加了缺失的 is_mm_prefix_lm 方法，该方法是 #40701 重构的一部分。