PR #36268 完整报告

vllm-project/vllm

[Audio] Bundle `get_generation_prompt()` params into `SpeechToTextParams`

合并时间：2026-04-22 12:24

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/36268

执行摘要

- 一句话：引入 SpeechToTextParams 数据类，统一语音转文本请求参数，简化模型接口签名。
- 推荐动作：该 PR 值得精读，特别是学习如何通过数据类统一接口参数的设计决策。关注 SpeechToTextParams 的定义和 build_stt_params() 的映射逻辑，这些是未来类似重构（如其他多模态接口）的参考。

功能与动机

基于与 @DarkLight1337 的讨论，目的是将 API 层请求字段和服务器端配置映射到 SpeechToTextParams 实例，从而避免在添加新模型特定标志时更改 get_generation_prompt() 的接口签名或任何现有模型实现。

实现拆解

1. 创建 SpeechToTextParams 数据类：在 vllm/config/speech_to_text.py 中定义 @dataclass，包含 audio、stt_config、model_config、language、task_type、request_prompt、to_language 等字段，作为所有语音转文本参数的统一容器。
2. 在协议层添加参数构建方法：在 vllm/entrypoints/openai/speech_to_text/protocol.py 的 TranscriptionRequest 和 TranslationRequest 类中添加 build_stt_params() 方法，该方法接收音频数据和服务器配置，返回 SpeechToTextParams 实例，将 API 请求字段映射到数据类属性。
3. 更新所有语音转文本模型：修改多个模型文件（如 cohere_asr.py、qwen3_omni_moe_thinker.py、qwen3_asr.py 等）中的 get_generation_prompt() 方法，将原先的多个参数替换为单个 stt_params: SpeechToTextParams 参数，方法内部从对象中解构所需字段，保持逻辑不变。
4. 导入和配置调整：更新相关文件的导入语句，添加 SpeechToTextParams 导入；同时为处理表单数据，在协议文件中添加对 vllm_xargs JSON 字符串的解析逻辑，并捕获解码错误以提供用户友好错误信息。
5. 文档配套：更新文档说明新参数对象的使用，但未涉及专门的新测试文件，依赖现有测试验证回归正确性。

关键文件：

- vllm/config/speech_to_text.py（模块 配置模块；类别 source；类型 core-logic；符号 SpeechToTextParams）：核心文件，定义了 SpeechToTextParams 数据类，作为所有语

音转文本参数的统一容器，是接口重构的基础。

- vllm/entrypoints/openai/speech_to_text/protocol.py（模块 入口点；类别 source；类型 dependency-wiring；符号 build_stt_params）：入口点文件，添加 build_stt_params() 方法，将 API 请求映射到 SpeechToTextParams，是参数传递的关键枢纽。
- vllm/model_executor/models/cohere_asr.py（模块 模型执行器；类别 source；类型 data-contract；符号 get_generation_prompt）：代表性模型文件，展示 get_generation_prompt() 如何适配 SpeechToTextParams，影响数据契约。
- vllm/model_executor/models/qwen3_omni_moe_thinker.py（模块 模型执行器；类别 source；类型 data-contract；符号 get_generation_prompt）：另一个关键模型文件，更新 get_generation_prompt() 以使用 SpeechToTextParams，体现跨模型的一致性变更。
- vllm/model_executor/models/qwen3_asr.py（模块 模型执行器；类别 source；类型 data-contract；符号 get_generation_prompt）：模型文件之一，类似更新 get_generation_prompt()，展示数据契约的广泛影响。

关键符号：get_generation_prompt, build_stt_params, SpeechToTextParams

关键源码片段

vllm/config/speech_to_text.py

核心文件，定义了 SpeechToTextParams 数据类，作为所有语音转文本参数的统一容器，是接口重构的基础。

```
from __future__ import annotations
from dataclasses import dataclass
from typing import TYPE_CHECKING

if TYPE_CHECKING:
    import numpy as np
    from vllm.config.model import ModelConfig

@dataclass
class SpeechToTextParams:
    """All parameters consumed by ``get_generation_prompt()``.

    ``TranscriptionRequest.build_stt_params()`` constructs this object,
    mapping API-level fields into typed attributes.  Models only receive
    this object, so new parameters can be added here without changing the
    ``get_generation_prompt`` signature.
    """
    audio: np.ndarray # 重采样后的音频波形，用于单个块
    stt_config: SpeechToTextConfig # 服务器级语音转文本配置
    model_config: ModelConfig # 模型配置
    language: str | None = None # ISO 639-1 语言代码（已验证 / 自动检测）
    task_type: str = "transcribe" # "transcribe" 或 "translate"
    request_prompt: str = "" # 可选文本提示，用于引导模型
    to_language: str | None = None # 翻译的目标语言（模型相关）
```

vllm/entrypoints/openai/speech_to_text/protocol.py

入口点文件，添加 build_stt_params() 方法，将 API 请求映射到 SpeechToTextParams，是参数传递的关键枢纽。

```
def build_stt_params(
    self,
    audio: "np.ndarray",
    stt_config: "SpeechToTextConfig",
    model_config: "ModelConfig",
    task_type: str,
) -> SpeechToTextParams:
    # 将 API 请求字段和服务器配置映射到 SpeechToTextParams 数据类
    return SpeechToTextParams(
        audio=audio,
        stt_config=stt_config,
        model_config=model_config,
        language=self.language, # 从请求中获取语言字段
        task_type=task_type,
        request_prompt=self.prompt, # 从请求中获取提示字段
        to_language=self.to_language, # 从请求中获取目标语言字段
    )

# 在 validate_transcription_request 方法中添加 vllm_xargs JSON 解析逻辑
xargs = data.get("vllm_xargs")
if isinstance(xargs, str):
    try:
        data["vllm_xargs"] = json.loads(xargs) # 将 JSON 字符串解析为字典
    except json.JSONDecodeError as e:
        raise VLLMValidationError(
            f"Failed to parse vllm_xargs. Must be valid JSON: {e}",
            parameter="vllm_xargs",
        ) from e # 捕获解析错误，提供用户友好的验证错误
```

vllm/model_executor/models/cohere_asr.py

代表性模型文件，展示 get_generation_prompt() 如何适配 SpeechToTextParams，影响数据契约。

```
@classmethod
def get_generation_prompt(cls, stt_params: SpeechToTextParams) -> PromptType:
    # 从 SpeechToTextParams 对象解构所需参数，避免直接传递多个独立参数
    audio = stt_params.audio
    stt_config = stt_params.stt_config
    language = stt_params.language
    request_prompt = stt_params.request_prompt

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

```
    # 原有提示生成逻辑保持不变，仅参数来源变更
    language_tag = f"<|{language}|><|{language}|>"
    pnc = True # TODO: 后续可配置化
    pnc_tag = "<|pnc|>" if pnc else "<|nopnc|>"
    default_prompt = (
        f"<|startofcontext|><|startoftranscript|>"
        f"<|emo:undefined|>{language_tag}{pnc_tag}"
        f"<|noitn|><|notimestamp|><|nodiarize|>"
    )
    prompt_text = request_prompt if request_prompt else default_prompt

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

评论区精华

- vllm_xargs JSON 解析错误处理：@gemini-code-assist[bot] 建议捕获 json.JSONDecodeError 并引发 VLLMValidationError 以提供用户友好的 400 级错误，此建议被采纳并实现。
- 抽象设计合理性：@DarkLight1337 询问是否应区分转录和翻译任务的逻辑，但最终结论是当前抽象足够，可以合并以解锁后续 PR（如添加 hotwords 参数）。
  - vllm_xargs JSON 解析错误处理 (correctness): 建议被采纳，PR 已添加异常捕获和 VLLMValidationError 抛出，确保错误处理正确。
  - 抽象设计是否区分转录和翻译任务 (design): 当前抽象被认为足够，可合并以解锁后续 PR（如添加 hotwords 参数），无需立即拆分。

风险与影响

- 风险：
  - 接口变更风险：所有语音转文本模型的 get_generation_prompt() 方法签名从多个参数改为单个 SpeechToTextParams 参数，若调用方未更新可能导致运行时错误；但 PR 已全面更新相关文件，风险较低。
  - JSON 解析安全：vllm_xargs 字段的 JSON 解析可能引入注入攻击或错误处理不当；PR 已添加异常捕获，但仍需确保输入验证和错误消息的适当性。
  - 测试覆盖不足：缺少对 SpeechToTextParams 数据类的专门单元测试，但现有测试（如 test_transcription_api_correctness.py）通过，表明回归风险可控。
- 影响：
  - 用户影响：无感知，因为 API 接口保持不变，仅内部参数传递方式优化，不影响现有请求格式。
  - 系统影响：提高代码可维护性和扩展性，未来添加新请求参数（如 hotwords）只需修改 SpeechToTextParams 而无需变更模型接口，减少代码变更量。

- 团队影响：开发者需熟悉新参数对象的设计模式，但遵循 TokenizeParams 的类似模式，易于上手和推广到其他模块。
- 风险标记：接口变更，JSON 解析错误处理，缺少单元测试

关联脉络

- PR #40460 [Bugfix] Pass effective chat template kwargs to reasoning parsers: 类似前端和核心接口修复，涉及参数传递优化，可对比学习设计模式。
- PR #40159 [MyPy] Enable mypy for vllm/model_executor/layers/: 同为重构类型，关注代码质量和维护性改进，体现团队对代码结构的持续优化。