

PR #44587 完整报告

vllm-project/vllm

[ASR] Add Long Audio benchmark and correctness test

合并时间: 2026-06-12 12:11

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/44587>

执行摘要

- 一句话: 新增长音频 ASR 基准测试与正确性测试
- 推荐动作: 值得阅读。该 PR 展示了如何为多模态任务设计轻量级基准数据集, 并围绕数据集异构性构建弹性加载逻辑。其中的 `load_data` 多分支设计和 `send_audio_file` 抽象可以作为类似场景的参考模式。维护者应关注安全评论中提到的路径验证建议, 考虑在后续 PR 中加固。

功能与动机

目前 ASR 正确性测试仅使用短音频样本, 无法覆盖长音频的 RMS 分块路径; RTFx 基准数据集过大 (18-100GB) 且样本短, 难以高效评估。因此需要引入轻量长音频数据集并添加相应测试。

实现拆解

1. 数据集扩展: 在 `ASRDataset` 类中新增 `EARNINGS22_CLEANED_DATASET` 和 `EARNINGS22_TINY_FILTERED_DATASET` 两个常量, 并扩展 `SUPPORTED_DATASET_PATHS`。重写 `load_data` 方法以根据数据集类型选择加载策略: 对于 `Earnings22-Cleaned-AA`, 通过 `snapshot_download` 下载本地 MP3 路径并映射为统一 schema; 对于 `Tiny-Filtered`, 禁用音频解码以避免 `torchcodec`。
2. 请求函数重构: 在 `async_request_openai_audio` 中提取 `send_audio_file` 内部函数, 支持直接传递 `BytesIO` 对象或文件路径, 同时保留原有数组转字节流的路径。新增 `audio_path` 键的支持, 使基准框架能够处理本地文件而不必依赖 HF 音频列为数组。
3. 正确性测试增强: 在 `test_transcription_api_correctness.py` 中添加 `load_audio_sample` 统一音频加载逻辑 (兼容 `array/path/bytes`), 修改 `transcribe_audio` 和 `bound_transcribe` 以传递 `extra_body` 参数。新增 `test_long_audio_wer_correctness` 参数化测试, 使用 `Earnings22-Cleaned-AA` 数据集, 并设置 `max_audio_len_s=1800` 绕过服务端 10 分钟限制。
4. 单元测试新增: 创建 `tests/benchmarks/test_audio_dataset.py`, 通过 `mock` 对象模拟 `ASRDataset.sample` 处理本地音频路径和嵌入音频字节的场景, 以及 `async_request_openai_audio` 处理本地文件路径的场景, 确保新分支得到验证。
5. 文档更新: 在 `docs/benchmarking/cli.md` 的可支持数据集中列出新添加的两个数据集名称。

关键文件:

- `vllm/benchmarks/datasets/datasets.py` (模块 数据集; 类别 `source`; 类型 `dependency-wiring`; 符号 `load_data`, `_disable_audio_decode`, `_materialize_local_audio_column`): 核心修改: 扩展 `ASRDataset` 以支持 `Earnings22` 数据集, 新增 `load_data` 分发逻辑、`_disable_audio_decode` 和 `_materialize_local_audio_column` 方法, 是数据集加载的主入口。
- `tests/entrypoints/speech_to_text/correctness/test_transcription_api_correctness.py` (模块 正确性测试; 类别 `test`; 类型 `test-coverage`; 符号 `load_audio_sample`, `transcribe_audio`, `bound_transcribe`, `process_dataset`): 长音频正确性测试: 新增 `test_long_audio_wer_correctness`, 使用 `Earnings22-Cleaned-AA` 数据集, 验证 RMS 分块路径的正确性; 同时重构了 `load_audio_sample` 等辅助函数以统一音频加载。
- `tests/benchmarks/test_audio_dataset.py` (模块 基准测试; 类别 `test`; 类型 `test-coverage`; 符号 `_ReadableBinary`, `read`, `_TokenizedPrompt`, `init`): 新增单元测试, 验证 `ASRDataset.sample` 和 `async_request_openai_audio` 对本地路径和嵌入字节的处理, 确保新分支得到覆盖。
- `vllm/benchmarks/lib/endpoint_request_func.py` (模块 请求函数; 类别 `source`; 类型 `core-logic`; 符号 `send_audio_file`): 重构核心请求函数: 提取 `send_audio_file` 内部函数, 支持通过文件路径发送音频, 同时保留对数组的支持, 是基准测试灵活性的关键。
- `docs/benchmarking/cli.md` (模块 文档; 类别 `docs`; 类型 `documentation`): 更新文档, 列出新增的 `ASR` 数据集, 方便用户了解可用的基准数据集。

关键符号: `load_data`, `_disable_audio_decode`, `_materialize_local_audio_column`, `send_audio_file`, `load_audio_sample`, `transcribe_audio`, `bound_transcribe`, `process_dataset`, `test_asr_dataset_sample_handles_local_audio_paths`, `test_asr_dataset_sample_handles_embedded_audio_bytes`, `test_async_request_openai_audio_handles_local_audio_paths`

关键源码片段

`vllm/benchmarks/datasets/datasets.py`

核心修改: 扩展 `ASRDataset` 以支持 `Earnings22` 数据集, 新增 `load_data` 分发逻辑、`_disable_audio_decode` 和 `_materialize_local_audio_column` 方法, 是数据集加载的主入口。

```
def load_data(self) -> None:
    # 根据数据集名称选择加载策略。
    # Earnings22-Cleaned-AA: 无 HF Audio 列, 需手动下载并构造音频路径。
    # Earnings22-Tiny-Filtered: 已有 HF Audio 列, 但禁用解码以避免 torchcodec。
    if self.hf_name == self.EARNINGS22_CLEANED_DATASET:
        self.data = load_dataset(
            self.dataset_path,
            name=self.dataset_subset,
            split=self.dataset_split,
            streaming=False,
            trust_remote_code=self.trust_remote_code,
        )
    if not getattr(self, 'disable_shuffle', False):
```

```

        self.data = self.data.shuffle(seed=self.random_seed)
        self._materialize_local_audio_column()
        return
    if self.hf_name == self.EARNINGS22_TINY_FILTERED_DATASET:
        super().load_data()
        self._disable_audio_decode()
        return
    super().load_data()

def _disable_audio_decode(self) -> None:
    # 将音频列设为 decode=False, 避免触发 torchcodec 解码
    from datasets import Audio
    self.data = self.data.cast_column('audio', Audio(decode=False))

def _materialize_local_audio_column(self) -> None:
    # 使用 snapshot_download 将数据集文件拉取到本地
    local_path_root = Path(
        hf_api().snapshot_download(self.hf_name, repo_type='dataset')
    )
    self.data = self.data.map(
        lambda item: {
            'audio': str(local_path_root / item['url']),
            'text': item['transcript'],
        }
    )
    self._disable_audio_decode()

```

tests/benchmarks/test_audio_dataset.py

新增单元测试, 验证 ASRDataset.sample 和 async_request_openai_audio 对本地路径和嵌入字节的处理, 确保新分支得到覆盖。

```

def test_asr_dataset_sample_handles_local_audio_paths(tmp_path: Path) -> None:
    # 生成一个极短测试音频文件
    audio_path = tmp_path / 'earnings.wav'
    _write_wav(audio_path, duration_s=0.1)

    # 构造一个仅含本地路径的虚拟数据集项
    dataset = object.__new__(datasets_module.ASRDataset)
    dataset.data = [
        {
            'audio': {
                'path': str(audio_path),
                'bytes': None,
            },
            'text': 'quarterly earnings call',
        }
    ]

    samples = dataset.sample(

```

```

tokenizer=_Tokenizer(),
num_requests=1,
output_len=32,
asr_min_audio_len_sec=0.0,
asr_max_audio_len_sec=1.0,
)

assert len(samples) == 1
# 检查 multi_modal_data 包含新添加的 audio_path 键
assert samples[0].multi_modal_data == {'audio_path': str(audio_path)}
# 确认 prompt 被正确构建
assert (
    samples[0].prompt
    == '<lstartoftranscript><len><ltranscribel><lnotimestamps>'
)

```

评论区精华

深度优先应用的自动化安全扫描在 [vllm/benchmarks/datasets/datasets.py:4072](#) 指出路径拼接风险: `item["url"]` 可能包含绝对路径导致任意文件读取, 建议验证解析路径是否仍在 `local_path_root` 下。虽然严重级别标注为 LOW, 但未在 PR 中得到明确修复后回应, 最终由维护者批准合并。另外, 作者在注释中解释了需要支持本地音频路径的原因: 某些数据集包含 MP3 等非 PCM 编码, 直接以文件形式发送可避免额外转码并控制数据大小。

- 路径遍历安全告警 (security): 未在 PR 中直接回复, 但 PR 最终被维护者批准合并, 问题未修复。

风险与影响

- 风险: 路径遍历风险: `_materialize_local_audio_column` 中 `Path(local_path_root) / item["url"]` 在获取外部数据集时存在潜在任意文件读取, 但目前数据集来源受控, 风险较低。
数据集可用性: 依赖 HuggingFace 数据集 `ArtificialAnalysis/Earnings22-Cleaned-AA` 和 `D4nt3/esb-datasets-earnings22-validation-tiny-filtered`, 若数据集下线或更改路径可能导致 CI 失败。测试超时: 长音频测试涉及 14-22 分钟的音频转录, 在 CI 环境下可能超时, 需合理配置超时限制。
- 影响: 该 PR 主要影响 ASR 测试与基准套件, 核心推理路径无变化。开发者和 CI 用户可以更高效地验证长音频转录的正确性与性能, 降低了基准测试的数据集获取成本 (从百 GB 降至 37 MB)。对于使用自定义数据集的用户, `async_request_openai_audio` 的新本地文件支持也提供了灵活性。影响范围限于测试与基准代码, 风险可控。
- 风险标记: 路径遍历告警, 第三方数据集依赖, 长音频测试超时, 缺少安全加固

关联脉络

- 暂无明显关联 PR