

PR #39793 完整报告

vllm-project/vllm

Bugfix: `use_existing_torch.py`: Glob recursive subdirs in requirements (fixes #39024)

合并时间: 2026-04-14 23:11

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/39793>

执行摘要

- 一句话: 修复 `use_existing_torch.py` 脚本无法递归处理嵌套 `requirements` 文件夹的问题。
- 推荐动作: 该 PR 变更简单直接, 适合快速浏览以了解 `requirements` 文件夹结构调整带来的影响。值得关注的是递归 `glob` 模式的使用方式, 以及如何保持向后兼容性。

功能与动机

PR body 明确指出, 由于 #39024 将 `requirements` 文件夹结构调整嵌套布局, 而 `use_existing_torch.py` 脚本原本假设所有 `requirements` 文件都在顶层目录中, 导致脚本无法处理嵌套的子目录文件。这破坏了脚本的核心功能——移除所有 `requirements` 文件中的 `torch` 依赖。

实现拆解

仅修改了 `use_existing_torch.py` 文件中的两行代码:

1. 将 `glob.glob('requirements/.txt')` 改为 `glob.glob('requirements//.txt', recursive=True)`
2. 将 `glob.glob('requirements/.in')` 改为 `glob.glob('requirements//.in', recursive=True)`
这使得 `glob` 模式能够递归匹配 `requirements` 目录及其所有子目录中的 `.txt` 和 `.in` 文件。

关键文件:

- `use_existing_torch.py` (模块 构建工具): 这是唯一被修改的文件, 包含了修复递归处理 `requirements` 文件夹的核心逻辑变更。

关键符号: `main`

评论区精华

review 讨论非常简短, 只有两个评论:

1. `gemini-code-assist[bot]` 的自动评论指出该 PR 更新了 `use_existing_torch.py` 以递归搜索 `requirements` 文件, 但没有需要评估的 review 评论。
 2. `ZJY0516` 简单批准并感谢贡献。没有出现技术争议或设计权衡讨论。
- 递归 `glob` 模式变更 (correctness): 通过启用 `recursive=True` 参数, 确保脚本能匹配所有层级的 `requirements` 文件。

风险与影响

- 风险：风险较低但需注意：
 1. 递归 glob 可能意外匹配到不应修改的文件，但考虑到 requirements 目录结构相对规范，风险可控。
 2. 脚本行为变更可能影响依赖该脚本的 CI/CD 流程，但 PR body 已说明这是为了恢复原有功能。
 3. 缺少测试验证递归 glob 的正确性，但变更逻辑简单直接。
- 影响：影响范围：
 1. 对用户：无直接影响，这是内部工具脚本的修复。
 2. 对系统：确保 use_existing_torch.py 能正确处理所有 requirements 文件，避免因未移除 torch 依赖导致的构建或测试问题。
 3. 对团队：修复了因文件夹结构调整引入的回归问题，恢复了脚本的预期行为。
- 风险标记：简单逻辑变更，缺少测试验证

关联脉络

- PR #39024 未提供，但根据 PR body 引用可知：PR body 明确提到 #39024 引入了 requirements 文件夹的嵌套结构调整，导致当前脚本失效，这是本次修复的直接原因。