PyPI 项目数据爬取和分析设计方案

王阳 2000010767

1 选题创意介绍

PyPI（Python Package Index）作为 Python 社区的核心资源库，包含了 50 多万个项目。然而，PyPI 的官网仅提供了简单的项目检索功能，并没有提供对其中数据进行精细的分析。本项目通过对 PyPI 中项目的编程语言、主题和维护者进行系统化的爬取和分析，可以很好地帮助我们了解开源社区的动态。具体而言，本项目有以下意义：

- **技术趋势分析：**通过分析各个项目所使用的编程语言，我们可以洞察当前哪些编程语言在 Python 生态系统中最受欢迎，从而对编程语言的技术趋势有所把握。
- **主题研究：**通过对不同主题下编程语言使用情况的分析，可以帮助我们理解在特定领域（如数据库、网络开发、操作系统等），哪些编程语言和技术框架是主流。
- **维护者网络：**分析项目的维护者信息，可以揭示出在开源社区中那些活跃的维护者以及他们之间的合作关系。这不仅有助于了解对开源社区贡献最大的个人或团队，还可以为新项目的协作提供网络基础。

2 数据获取

2.1 抓取时长

由于 PyPI 包含 50 多万个项目，单线程逐一抓取将耗费大量时间。为了提高抓取效率，我们将采用多线程技术进行并行抓取。预计在 2-3 天内完成全部数据的获取。具体时间会根据服务器响应速度和网络条件有所浮动。

2.2 数据规模

在本次项目中，我们计划从 PyPI 爬取以下关键信息：

- **项目名称**：每个项目的唯一标识符和主要描述。
- **维护者列表**：项目的所有维护者名称。

- **所使用编程语言列表**：项目中使用的所有编程语言。
- **主题列表**：项目所涉及的所有主题。

3 数据分析、处理、保存

3.1 数据存储

由于爬取数据超过 50 万条，为了便于管理和高效查询，我们将使用 SQLite 数据库存储爬取的数据。

3.2 数据分析和处理

- **编程语言使用比例**：我们将基于项目所使用的编程语言，统计不同主题下各个语言的使用比例。通过对数据库的查询和聚合操作，可以快速得到不同主题下的编程语言分布情况。
- **活跃的维护者**：分析每个维护者参与的项目数量，识别出那些在社区中最为活跃的维护者。进一步，我们将分析这些维护者之间的合作关系，构建维护者之间的合作网络图。
- **主题分布**：统计分析各个主题下的项目数量和特征，揭示哪些主题在当前社区中最受关注。

3.3 数据展示

为了让分析结果更加直观，我们将使用 pyecharts 库制作网页，展示上述分析结果。主要包括以下几部分内容：

- **直方图**：展示不同主题下各个编程语言的使用比例。每个条形图代表一个主题，其中不同颜色的条状表示不同的编程语言，条状的长度代表其使用比例。
- **词云**：展示维护项目最多的那些维护者。词云图通过字体大小和颜色深浅展示维护者的活跃程度，越活跃的维护者名字越大、颜色越深。
- **关系图**：展示维护者之间的合作关系。通过图节点和连线表示维护者及其合作关系，节点大小表示维护者活跃度，连线表示他们之间合作项目的数量。