

南京大學

計算機科學與技術系

軟件需求工程實驗報告

實驗名稱：需求的抽取與分類

學 號：

姓 名：

指導教師：

實驗地點：

實驗時間：2022 年 10 月

目 录

- 1. 引言2
 - 1.1 成员与分配2
 - 1.2 实验名称2
- 2. 实验概述2
 - 2.1 实验背景2
 - 2.2 实验目的2
 - 2.3 实验主题2
 - 2.4 实验任务2
- 3. 实验内容2
 - 3.1 实验对象及介绍2
 - 3.2 数据来源3
 - 3.3 实验方法3
 - 3.4 实验工具3
- 4. 实验结果及分析3
 - 4.1 实验结果3
 - 4.2 需求分类9
 - 4.3 需求分析11
- 5. 实验结论与扩展14
- 6. 参考文献15

一、 引言

1. 成员及分配比例

成员	任务分配	比例
	数据爬取与处理	1/3
	数据分类	1/3
	报告撰写	1/3

2. 实验名称
需求的抽取与分类

二、 实验概述

1. 实验背景
- 软件需求分析属于软件生命周期的第一阶段，直接关系着软件的成功与否。需求分析由软件分析人员与用户共同完成，对开源项目来说，用户无处不在，可以通过多种渠道获取软件相关信息，提取其中潜在的需求。对需求进行分类，有助于进一步精化软件的作用范围，明确系统必须完成的功能，对目标系统 提出完整、准确、清晰、具体的要求。
2. 实验目的
- 抽取软件需求并进行分类
3. 实验主题
- Smart IDE
4. 实验任务
- 选定一个开源 IDE 项目并确定需求的信息来源，收集有效数据，使用特定技术提取潜在的需求，对需求进行分类。

三、 实验内容

1. 实验对象及介绍
- Visual Studio Code（简称 VS Code）是一款由微软开发且跨平台的免费源代码编辑器。该软件支持语法高亮、代码自动补全（又称 IntelliSense）、代码重构功能，并且内置了命令行工具和 Git 版本控制系统。用户可以更改主题和键盘快捷方式实现个性化设置，也可以通过内置的扩展程序商店安装扩展以拓展软件功能。
- VS Code 使用 Monaco Editor 作为其底层的代码编辑器。
- Visual Studio Code 的源代码以 MIT 许可证在 GitHub 上释出，而可执行文件使用了专门的许可证。

微软在 2015 年 4 月 29 日举办的 Build 2015 大会上公布了 Visual Studio Code 的开发计划；同日，其预览版本发布。2015 年 11 月 18 日，Visual Studio Code 在 GitHub 上开源，同时宣布将支持扩展功能。2016 年 4 月 14 日，Visual Studio Code 正式版发布。

在 2019 年的 Stack Overflow 组织的开发者调查中，Visual Studio Code 被认为是最受开发者欢迎的开发环境。据调查，87317 名受访者中有 50.7% 的受访者声称正在使用 Visual Studio Code。（参考文献 1）

2. 信息来源

本次实验我们小组选择对开源软件 VScode 进行研究与分析。

在进行充分的考察与选择后，我们选择对 VScode 的 Github 仓库中的 issue 与 PR 进行分析。（参考文献 2）

在考察中，我们发现，对于 VScode 项目，截至调查的一刻，其共有 305 页，计 7624 个 open 的 Issues，同时还有 5465 页，计 136602 个 Closed 的 Issues。其中 open 的 Issues 中，最近的活动在调查前的十五分钟。同时，截至调查的一刻，其共有 14 页，计 341 个 open 的 PR，同时还有 641 页，计 16009 个 Closed 的 PR，其中 open 的 PR 中，最近的活动在调查前的八小时。综合以上数据，我们认定，VScode 在 Github 上的 Issue 与 PR 栏目是仍然活跃的，是具有一定的研究与参考价值的，可以作为本次实验的信息来源。

3. 实验方法

显然地，对于 VScode 在 GitHub 上数以万计的 Issue 与 PR，通过人工比对来获取需求的想法是十分不明智的选择，因此，通过小组讨论，我们最终决定通过爬虫技术进行需求的获取。由于对爬虫技术的不熟悉，我们决定将重点放在数据的获取上，首先进行爬虫技术的学习。通过不懈努力与辛勤实验，虽然我们使用的爬虫技术仍然存在相当的问题与瑕疵，但我们最终成功获取了所需要的结果。

4. 实验工具

NLTK 自然语言处理工具

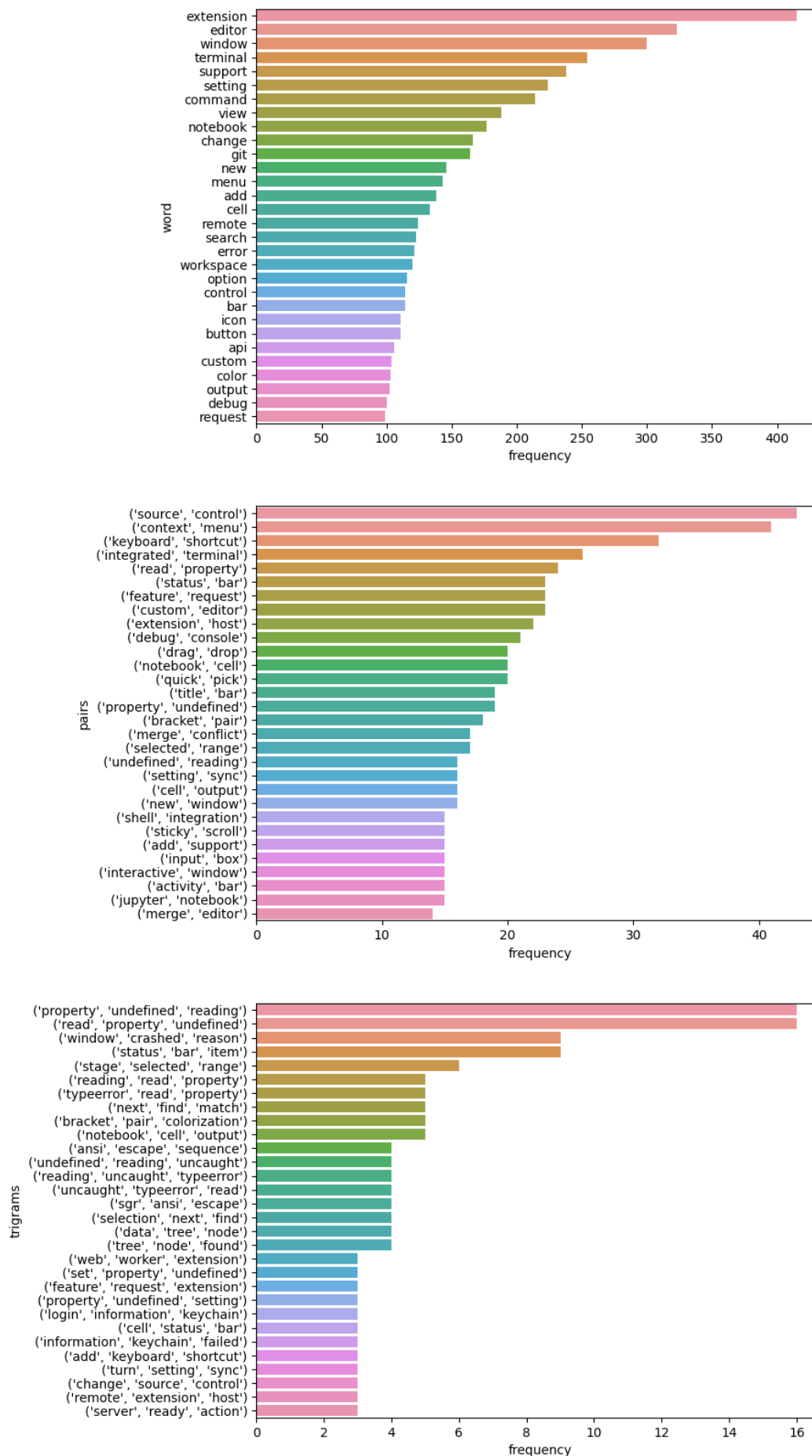
通过一定时间的了解与研究，我们认为单单使用简单的爬虫工具并不能满足我们的需求，因此，我们使用了 NLTK 自然语言处理工具对爬取时的数据进行分段，由于爬取的数据都为英文格式，因此正是 NLTK 发挥功用是时候。通过 NLTK 自然语言处理工具的帮助，我们通过分段处理获得了相关数据。

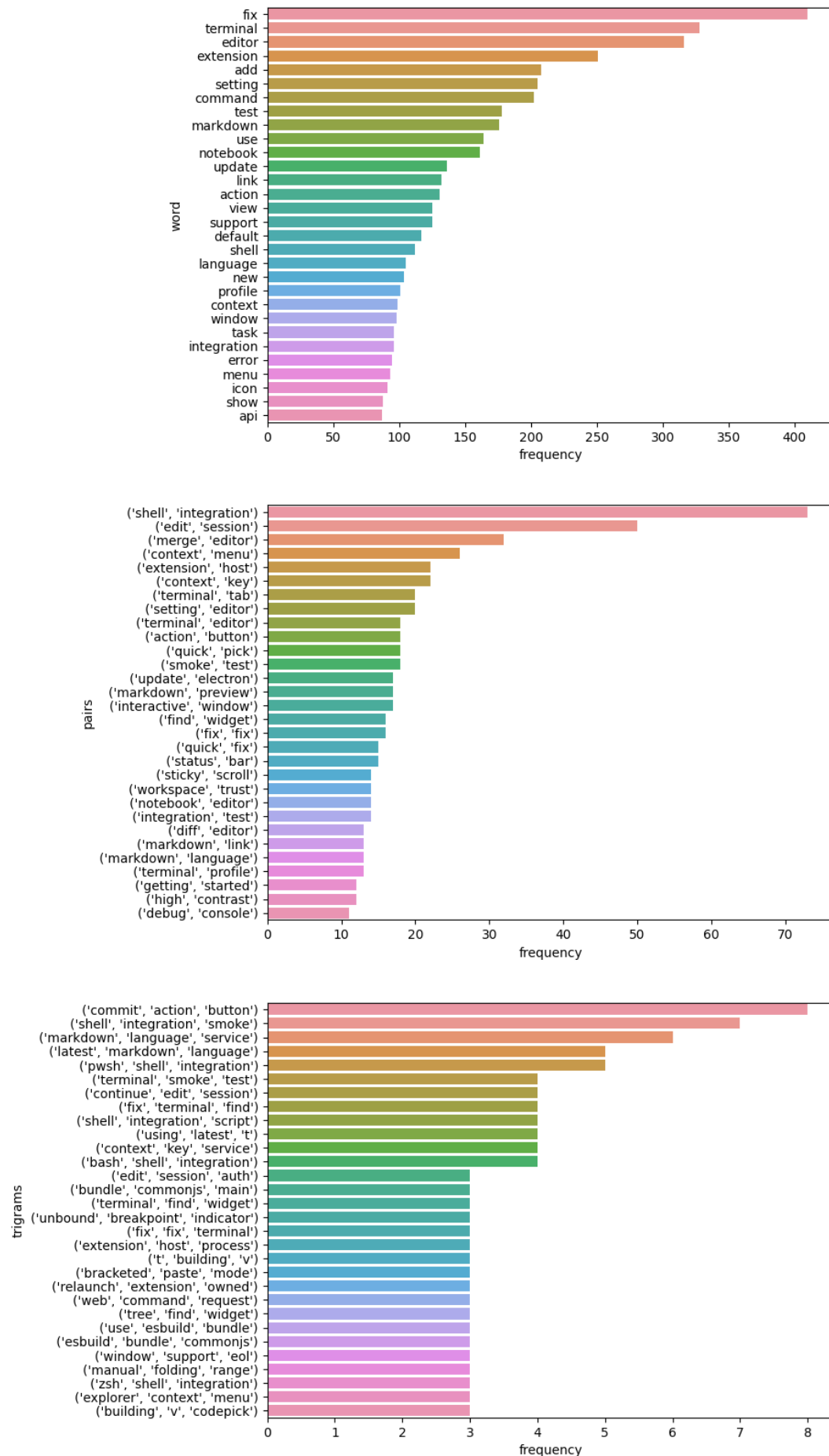
四、 实验结果及分析

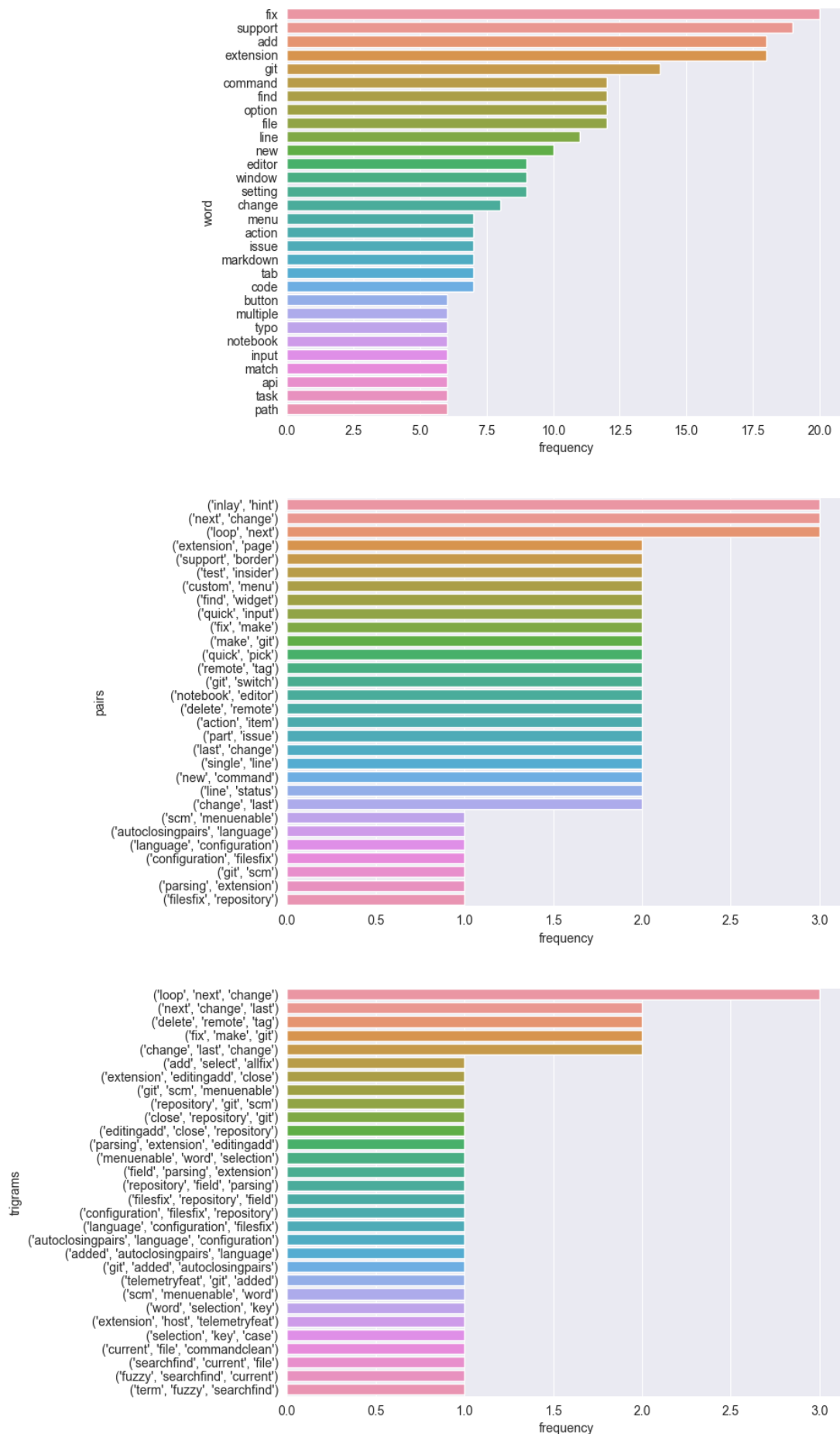
1. 实验结果

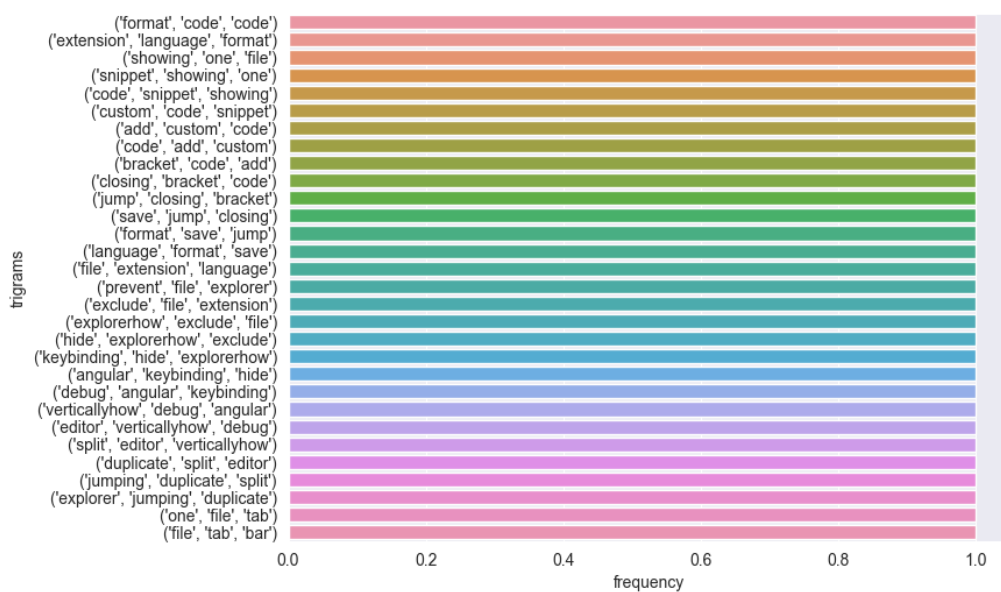
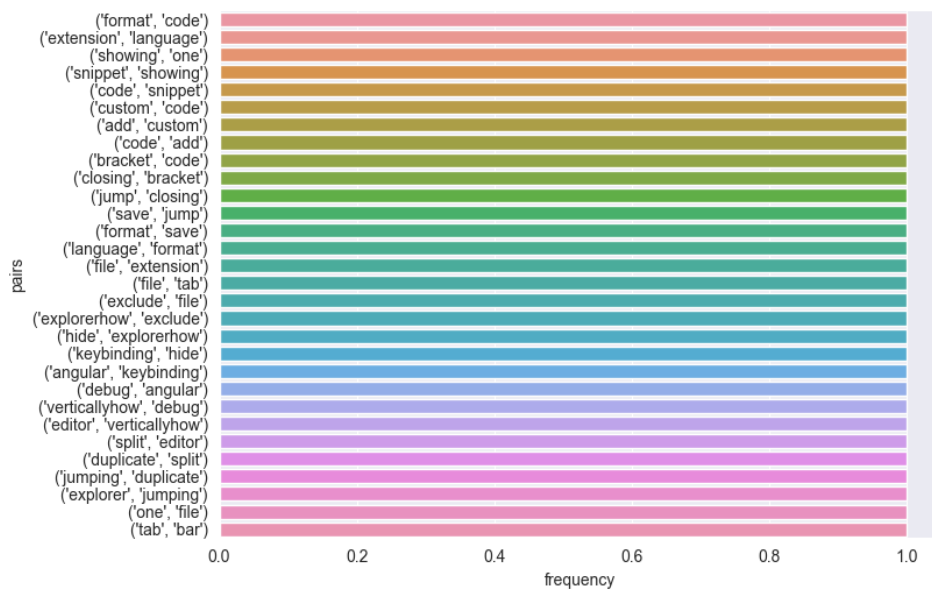
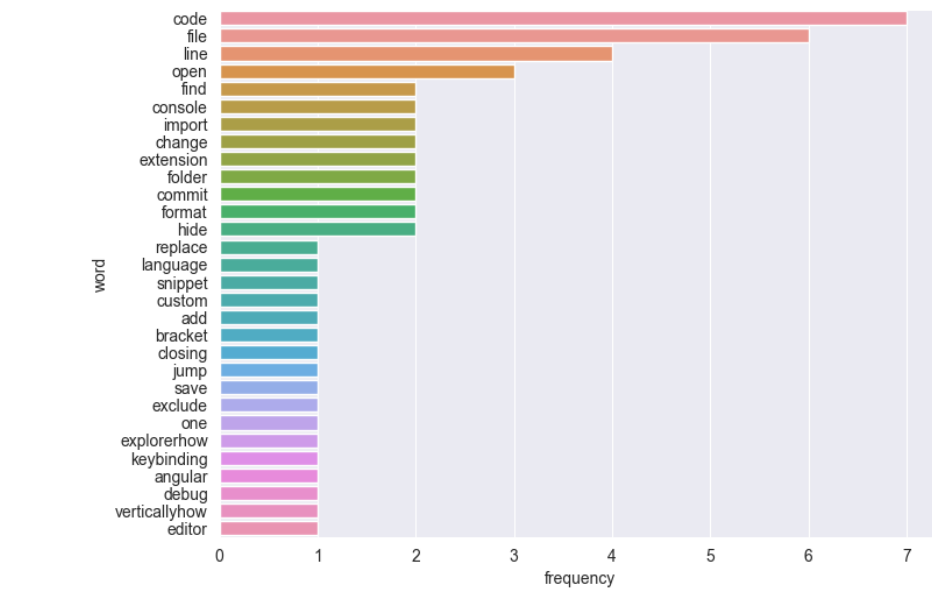
首先附上我们项目所使用的 GitHub 仓库链接：<https://github.com/raindrops-0199/vscodeAnalyze>

我们通过爬虫技术将 GitHub 上 VScode 项目的全部 open 的 Issue 爬取到本地，计 7624 个文件，再通过自然语言处理工具对爬取到的文件进行词汇分析，剔除掉一些统概性过高的词汇，我们整理了出现频率高且蕴含具体需求的词汇，将之利用 python 的 matplotlib 库绘制了可视化图形文件，并将之转化为 png 图片，四幅可视化词频图按顺序分别为 Issue 词频图，PRopen 与 PRclosed 词频图，Stackoverflow 词频图：



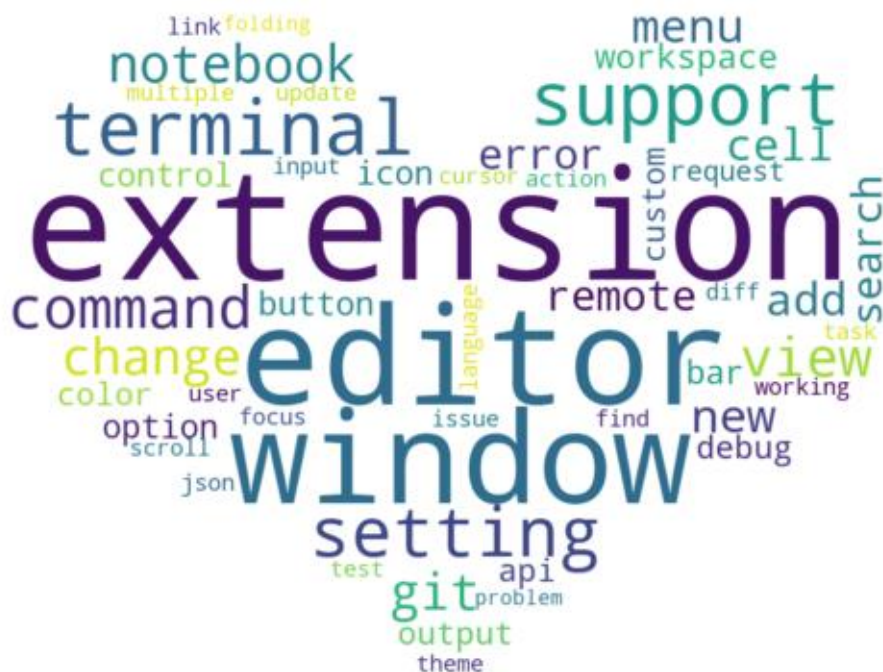






同时，我们在进行词频统计后对关键词进行了进一步的分析与处理，最终得出了以下四幅词云分布图：

Issue 词云图：

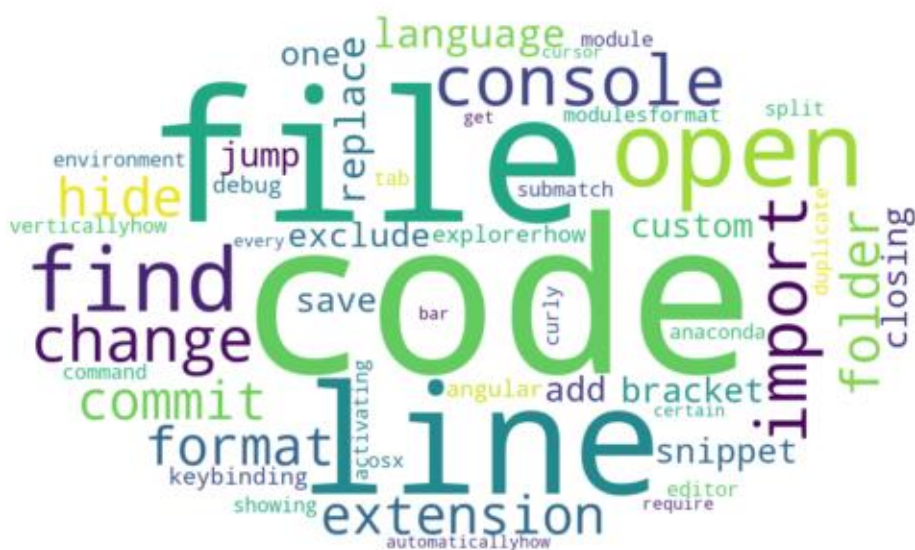


PR 词云图：





Stackoverflow 词云图:



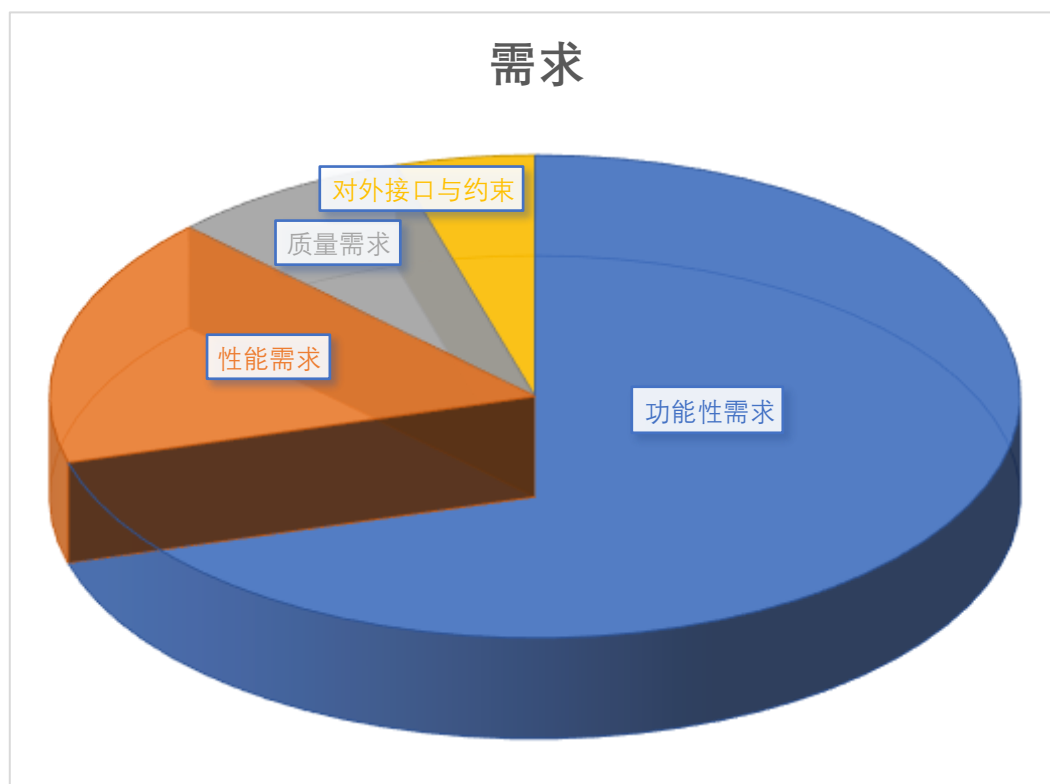
2. 需求分类

首先对于软件需求而言，软件需求分为功能需求与非功能需求，其中非功能需求又分为性能需求，质量属性，对外接口与约束。其中，功能需求体现在系统与用户之间的交互，帮助用户解决问题，完成任务。性能需求指软件的速度，容量，并发性等。质量需求包括可靠性，可维护性，安全性等。对外接口顾名思义，为接口的名称与用途等。约束指系统开发环境，涉及的法律信息条文等。

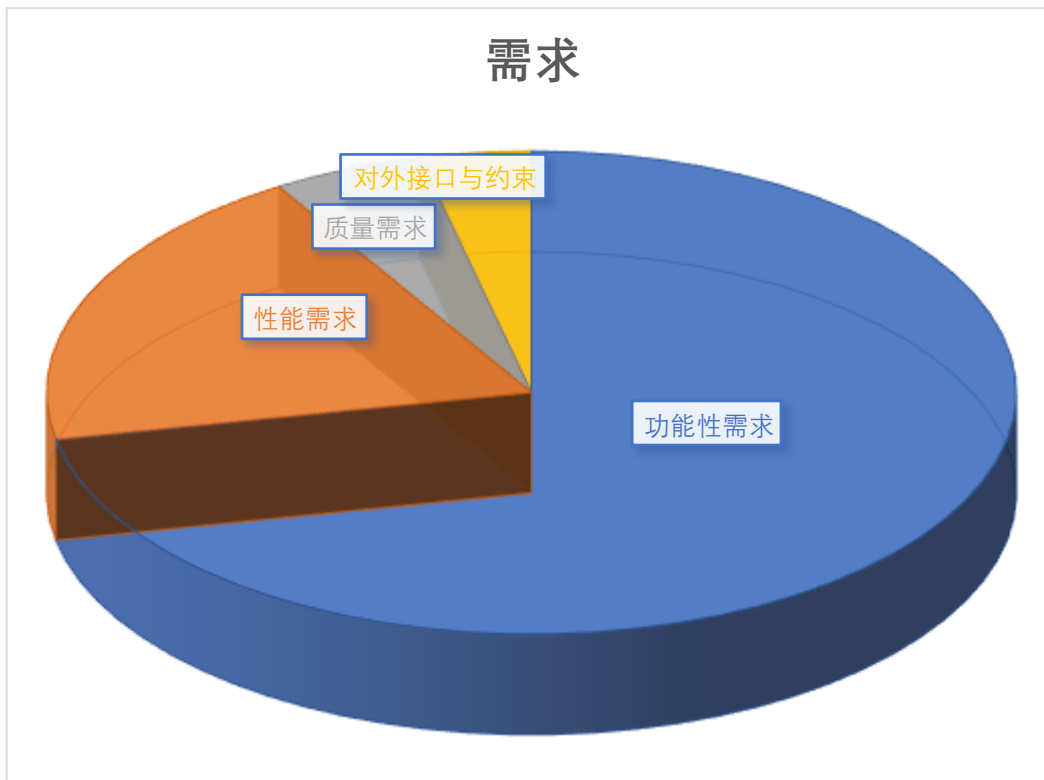
综合上述概念与我们的实验结果，我们将结果进行了分类，分类结果如下：

在对 Issue 栏目中 7608 条数据在筛出掉部分无价值以及价值较低的数据后进行分析，功能性需求包括但不限于：299 条与用户以及其使用过程相关的词条，35 条关于用户引导以及帮助的词条，30 条系统反馈相关的词条，355 条建立与完成任务相关的词条。性能需求中最多

的是 148 条关于软件运行速度与流畅性的词条。质量需求中有：85 条关于软件崩溃的问题，5 条关于软件安全性的需求。对外接口相关的词条有 17 条，约束相关的词条有 28 条。综合全部数据后可以得出：



在对于 PR 的 8289 条数据在筛出掉部分无价值以及价值较低的数据后进行分析，功能性需求包括但不限于：329 条与用户以及其使用过程相关的词条，36 条关于用户引导以及帮助的词条，11 条系统反馈相关的词条，323 条建立与完成任务相关的词条。性能需求中最多的仍然是关于软件运行速度与流畅性的词条，有 104 条。质量需求部分词条有 15 条。对外接口相关的词条有 8 条，约束相关的词条有 19 条。综合全部数据后可以得出：

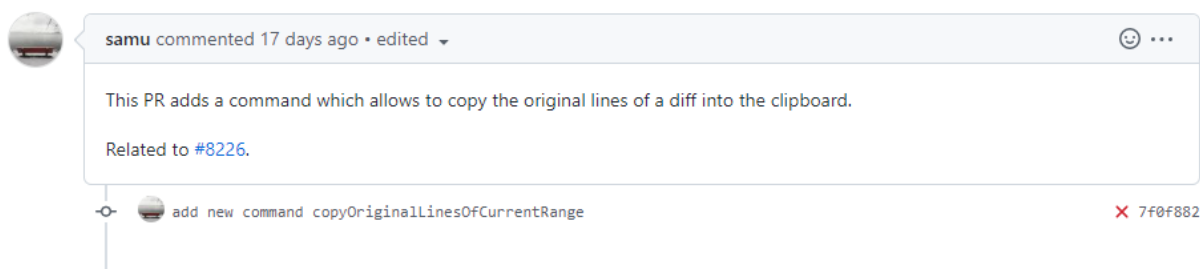


通过对与 Issues 与 PR 的大量数据分析可以发现，用户需求主要集中在功能性需求方面，对于对外接口与约束方面的需求较少。结合 VScode 软件本身的性质以及用户群体的使用需要来看：VScode 作为一款编译软件，用户群体大多为程序员群体，因此对于软件交互等方面更为重视；并且 VScode 的界面功能相对较多，对于用户的引导与帮助也至关重要。这应该也是功能性需求占据大半的原因之一。而非功能需求方面，用户对于性能需求也最为重视，做为程序员，给自己程序 debug 已然不易，因此也不希望编译软件不时地卡顿给本就烦躁的心情火上浇油。质量需求不高说明了软件本身的稳定性以及可靠性能够满足大多数用户的需求，如果软件动不动就崩溃导致辛苦码的代码未被保存的话，质量需求的数量断然不会太低。

3. 需求分析

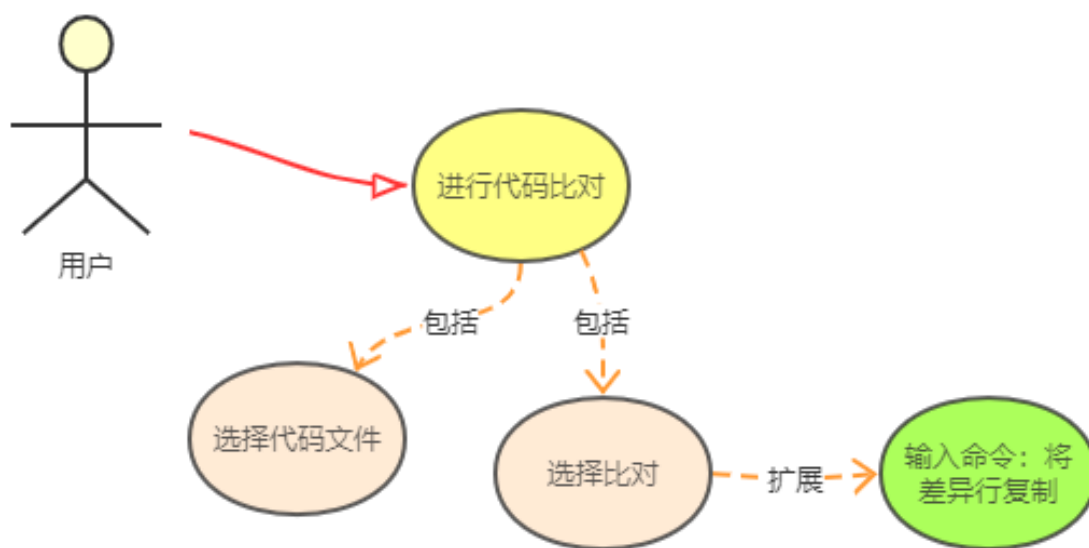
接下来我们从已经分好类的需求中抽取一部分，并在 Issue 栏目中找到了具体的实例对需求进行具体的分析，分析结果如下：

功能需求类：



该 PR 添加了一个命令，该命令允许用户在文件对比过程中将有差异的代码复制到剪贴板中，便于后续修改与使用。

分析：首先通过用例图查看：



可以看到，用户原本的操作为选择代码文件并进行比对，而该需求将扩展比对时可进行的操作，用户可以输入命令并进行差异行的复制。

显然，新添加的功能在紧迫性上分析并非十分紧迫，优先级自然不必很高。但是，从用户体验与软件功能性方面来讲，该功能丰富了用户体验，完善了产品功能，具有一定的开发价值。



bersbersbers commented 11 minutes ago

When updating an image file in a repository, it is often unhelpful to have two images side-by-side. If visible differences do exist, they are often hard to spot. And there isn't even an indication if there is *any* visible difference between the files, or whether differences are only in the metadata.

GitHub has (had) the same issue until nicely solved it, see <https://github.com/bentleyvk/github-pr-image-comparer> and <https://chrome.google.com/webstore/detail/github-pr-images-comparer/fdbkdbjkkjceenjnkdpjiiibkhlknlm> github-pr-images-comparer

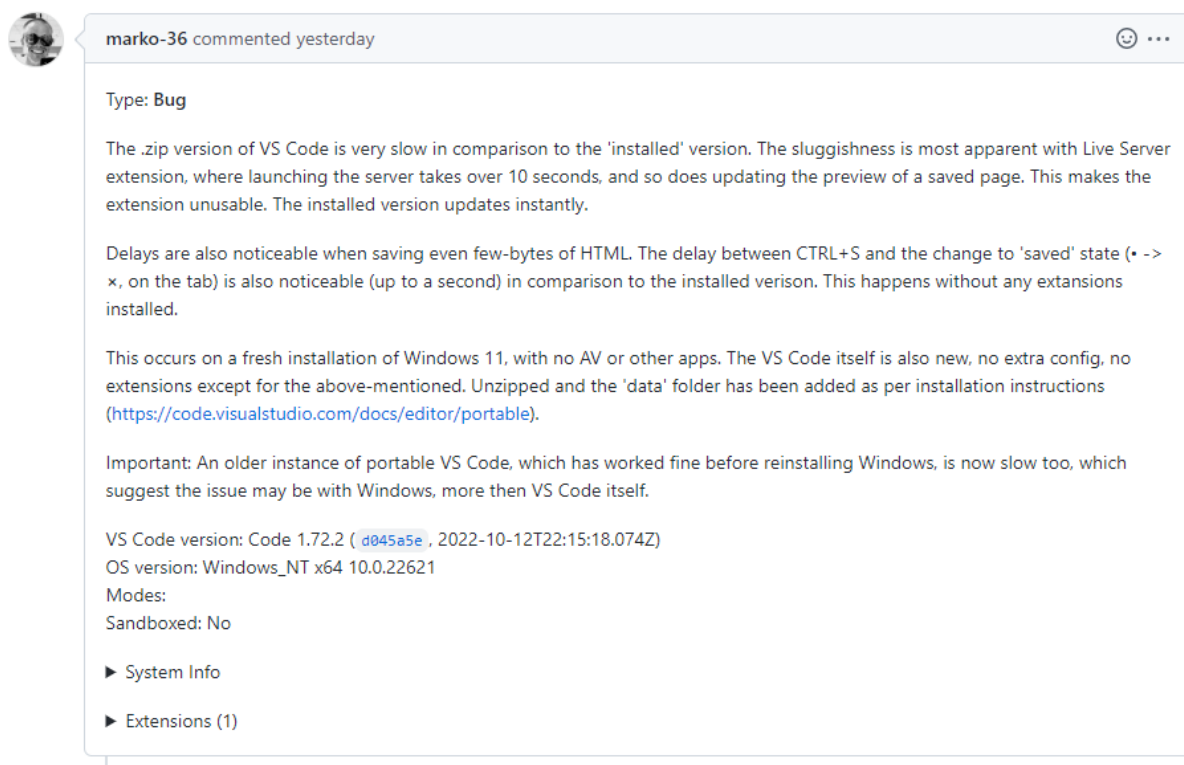
It would be great if VS Code could offer similar possibilities. I would not mind it being a separate extension and might even contribute to that, but I am first opening this issue to see how big user interest would be anyway.

Related: [microsoft/vscode-pull-request-github#2918](https://github.com/microsoft/vscode-pull-request-github/issues/2918)

该用户发布了一个需求：在更新文件时添加图像像素比对功能，通过像素比对图像文件的相似性，进行图像文件的去重。

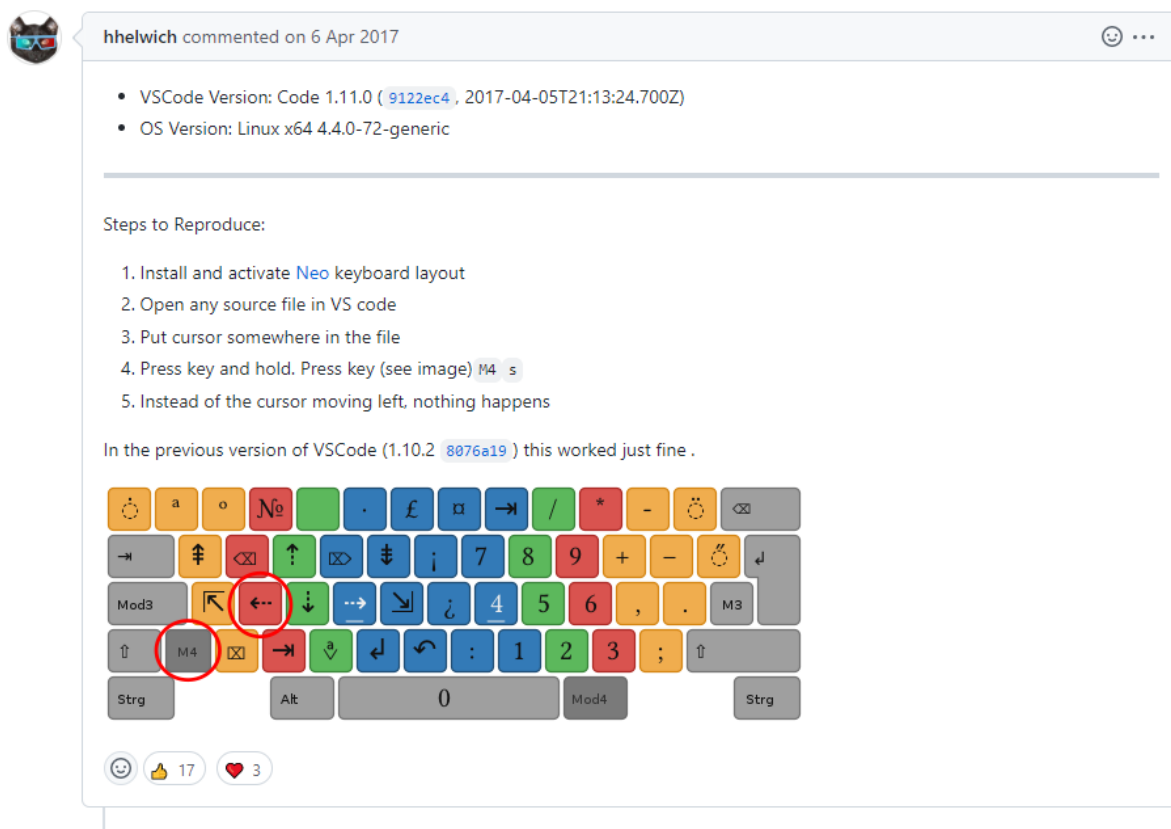
分析：vscode 不仅可以保存代码文件，图像文件也是保存的一部分，同时，从需求中我们可以看到，Github 已然解决了此类需求。因此，综合上述研究，可以认定，像素去重需求可行性存在，优先度高，紧迫性高，由于存在现有的成果，难度并不大，值得开发此类功能。

非功能需求：



用户需求：VScode 在 Windows11 版本中可移植的.zip 包版本与 install 版本相比运行速度极慢，需要排查原因并对其进行速度上的优化。

分析：作为发行版软件，VScode 应当适配各大主流操作系统，尤其是新出现的 Windows11 版本。同时，相比 install 版本，.zip 版本也因当能够正常运行，而不是数十秒的运行速度差距，因此，对该需求进行分析，我们可以认定，该需求可行性存在，优先度高，紧迫性高，对于 VScode 在 Windows11 上的用户体验至关重要，需要立刻着手优化。



用户需求：在 Neo 键盘布局中，在某些具体的操作下，必然会出现键盘按键失灵的情况，该情况是完全可复现的。

分析：该键盘布局具有相当数量的用户，则应当对该键盘的操作进行处理，而且，由于键盘布局格式的不可替代性，对于该需求处理的优先度高，紧迫性高，需要进行处理。

五、实验结论与拓展

1. 需求提供的重要性

通过此次实验，我们发现了 VScode 应用在 GitHub 上 Issue 与 PR 内容对于 VScode 整体项目进步与完善的重要性。可以发现，其 Issue 与 PR 项目栏目是十分活跃的。其中 GitHub 用户的活跃度是功不可没的。同时作为 VScode 项目发言人，VSCodeTriageBot 的回应也是相当迅速的。总体来看，其 Issue 栏目是由用户与项目方共同繁荣的。相信这也是 VScode 作为开源软件能够如此成功的原因之一。

作为拓展，我们想知道一个充分繁荣的需求提供平台对项目的成功是否存在一定的帮助。因此，我们又查询了一些著名的开源项目在 GitHub 上的情况。对于 vue 项目在 GitHub 中的相关情况。截至查询前一刻，vue 项目在 GitHub 共有 9902 个 open 或 closed 的 Issue，共 2314 个 open 或 closed 的 PR。对于 react 项目在 GitHub 中的相关情况。截至查询前一刻，react 项目在 GitHub 共有 11796 个 open 或 closed 的 Issue，共 12825 个 open 或 closed 的 PR。对于 Vim 项目在 GitHub 中的相关情况。截至查询前一刻，Vim 项目在 GitHub 共有 6752 个 open 或 closed 的 Issue，共 4688 个 open 或 closed 的 PR。

显然，对于我们抽取的三个比较知名的开源项目，其在 GitHub 上的需求征询部分都是具有相当数量的。我们相信，这些需求必然会对这些项目的完善做出或多或少的贡献。以 Vue 为例，关注尤雨溪（vue 项目负责人）的主页，可以看到他是十分鼓励人们在其项目的 Issue

栏目中提出疑惑并热心解答的。毕竟，需求促进进步。

同时，作为扩展，我们在参考资料之外又找到了一些知名项目的需求征询主页，浏览主页，我们惊叹于用户的想法之深刻与项目方回应之迅速。对于部分我们一直使用的项目，我们甚至看到了已经实装的功能，这必然是需求推动的产品迭代。作为分享，我们将一些开源项目的主页置于文末参考文献中。

2. 需求分析的必要性

本实验小组成员都曾经在校级组织南小宝公司学习，都曾经经历过南小宝软件功能迭代。因此，我们深刻地认识到了需求分析的必要性。结合个人经历，软件的开发绝对不是产品经理拍脑袋想出来的。而是要经过产品，运营，研发等多个团队在一起经过多轮会议研讨最终成型的。其中，产品的功能还要经过对用户的需求调研后才能够最终进入研发阶段。即便经过了如此多轮的开会讨论，产品调研，南小宝软件仍然开发出了一些鸡肋的功能，例如蓝鲸笔记等一些上线半年仅几十的活跃度的功能。现在想来，如果能够做好更加细致的软件需求分析，那么当时在选择功能迭代的开发中，也许我们会少走很多的弯路吧。

3. 总结

很显然，成熟的软件都是具有良好的团队对其进行细致的软件需求分析，最终开发团队才能够针对性的对用户真正需要的功能做出研发与改进。本次实验带给我们的感触良多。做一个长期运营的优秀的成功的软件绝对不是一日之功，是需要进行详细的事前分析，用户需求统计，良好的软件更新与迭代与维护，才能够做好一个优秀软件的最基础操作。而在其中，软件需求分析是一个优秀软件的基础，同时也是其开发与迭代的最明亮的导向标。通过此次实验，我们真正明白与懂得了软件需求分析的重要程度，相信这会对我们未来的学习与工作有着极好的教育意义。

六、 参考文献

- [1].[Wiki VScode 介绍](#)
- [2].[VScode 在 GitHub 上的源码仓库](#)
- [3].[Vim 项目官方交流邮件地址](#)
- [4].[Visual Studio 博客及文档](#)
- [5].[Vue 在 Dev 的社区](#)
- [6]. 我们项目所使用的 GitHub 仓库链接: <https://github.com/raindrops-0199/vscodeAnalyze>

实验评分: _____

指导教师签字: _____

2022 年 10 月