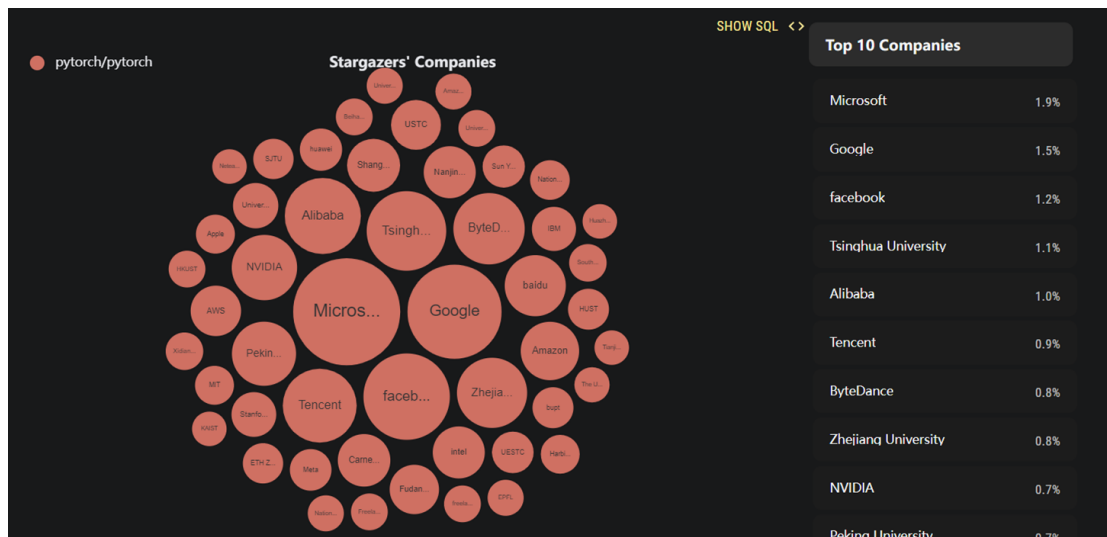


需求研讨会记录

主要分析的项目 PyTorch

- 数据存储：
 - 存在服务器的数据库中（本地）
 - 需要显示当前数据的截止时间
- 数据更新：
 - 理论上是一键更新（用户手动点击更新按钮时，才执行更新操作）
 - 可以不实时返回更新结果，只给一个预计的更新完成时间
- 对于贡献者身份的界定：
 - 个人代码量为所有贡献者中的前20%
 - 或
 - 这一堆人的总代码量达到项目的80%
（取人少的那一种方式，或者放飞自我）
- 需要统计的数据
 - 贡献者的人数&公司信息随时间的变化
 - 项目的 request/commits/issues 数量随时间的变化（应该是每个时间间隔里增量的变化）
 - 可以以month为大时间跨度，week为小时间跨度进行衡量（也可以diy）
 - issue从open到close的时间跨度（主要体现社区活跃程度）
- 关于跨项目对比
 - 主要实现两两对比（就是说还要可视化一个和PyTorch相当的项目数据）
 - 对于**所有可视化数据**进行横向比较（可以支持diy比较项目，也可以不支持）
- 如何衡量社区发展速度（两个维度）
 - 单位时间间隔内 pull request/commits/issues 的总量
 - pull request/commits/issues 参与成员的数量变化（增或减）
- UI（**感觉老师只对公司信息有很强的执念**）
 - 老师希望一个**低饱和度**的配色方案
 - 公司信息（老师的意思是一坨圆圈 + Top10 List）
因为涉及公司很多，圆圈之间不好辨别，最好加一个 list



- 讨论数据分类
 - 过程
 - 将 pull request 分为 design 相关/design 无关的两类
 - 可以对所有 design 相关的讨论进行词频统计，然后进行词云显示之类的
 - 下面表格的 category & description 内容可以自定义

Category	Occurrences	Description
	Agreed on	
code	11	Implementation issues
maintainability	14	Future plans, OS support, code standards...
testing	1	Tests and testability
robustness	13	Robustness, safety, security
performance	2	Performance, runtime optimization
configuration	4	Configuration files, flags and options
documentation	1	Documentation in-code and off-code
clarification	7	Generic question

- pull request 中的“开发者角色”界定

老师的意思是让我们自己对用户分类（可以只划分核心开发者/非核心开发者）

时间	
开发者名称	
开发者角色	