

开发人员在 Gitter 上的实时聊天初探

林石

shilin@iscas.ac.cn
中国科学院软件研究所, 中国科学院大学, 中国

小陈

chenxiao2021@iscas.ac.cn
中国科学院软件研究所, 中国科学院大学, 中国

Ye Yang

yyang4@stevens.edu
史蒂文斯理工学院系统与企业学院 美国新泽西州霍博肯

姜涵之姜子友

{hanzhi2021, ziyou2019}@iscas.ac.cn
中国科学院软件研究所, 中国科学院大学, 中国

牛楠

nan.niu@uc.edu
辛辛那提大学EECS系, 美国俄亥俄州辛辛那提

王青

wq@iscas.ac.cn
中国科学院软件研究所计算机科学国家重点实验室, 中国科学院大学, 中国

抽象的

Gitter 和 Slack 等现代通信平台在支持软件团队合作方面发挥着越来越重要的作用, 尤其是在开源开发方面。此类平台上的对话通常包含密集的、有价值的信息, 可用于更好地理解 OSS 开发人员的沟通和协作。然而, 这方面的工作很少。为了弥补这一差距, 本文报告了对开发人员实时聊天的第一次全面实证研究, 调查了他们何时互动、社区结构是什么样的、讨论了哪些主题以及他们如何互动。我们在第一阶段手动分析 749 个对话, 然后在第二阶段自动分析超过 173K 对话。我们发现开发人员倾向于在工作日进行十次以上的交谈, 尤其是在周三和周四 (UTC), 观察到三种常见的社区结构, 开发人员倾向于讨论 API 使用和错误等话题, 以及六种对话交互模式在实时聊天社区中被识别。根据研究结果, 我们为个人开发人员和 OSS 社区提供建议, 突出平台供应商所需的功能, 并阐明未来的研究方向。我们相信, 这些发现和见解将有助于更好地了解开发人员的实时聊天, 为其他研究人员铺平道路, 以及更好地利用和挖掘嵌入在海量聊天历史中的知识。

关键词

实时聊天、团队交流、开源、实证研究

ACM 参考格式:

林石、萧晨、叶阳、姜寒之、姜子友、南牛、王青。2021。初看开发者在 Gitter 上的实时聊天。在第 29 届 ACM 联合欧洲软件工程会议和软件工程基础研讨会 (ESEC/FSE 21) 上, 2021 年 8 月 23 日至 28 日, 希腊雅典。ACM, 纽约, 纽约, 美国, 13 页。 <https://doi.org/10.1145/3468264.3468562>

1 简介

Gitter、Slack、Microsoft Teams、Google Hangout 和 Freenode 等在线交流平台比以往任何时候都在团队交流和协作中发挥着重要作用。作为

作为开发人员社区之间同步文本通信的一种类型, 实时聊天允许开发人员接收其他人的实时响应, 在某些情况下取代异步通信, 如电子邮件 [40, 58, 59]。对于由全球分布的开发人员贡献的开源项目以及许多因 COVID-19 大流行而允许开发人员在家工作的公司而言尤其如此。来自在线交流平台的对话包含用于研究开发人员行为的丰富信息。图 1 举例说明了来自 DeepLearning4j Gitter 社区的一段实时聊天日志。每个话语都由时间戳、开发者 ID 和文本消息组成。此外, 聊天日志中嵌入了两个对话框。第一个是报告有关“earlystop”的问题, 第二个是请求文档支持。因此, 有价值的信息, 如

CCS 概念

·软件及其工程→开源模型; 一般和参考→实证研究。

通讯作者。

允许为个人或课堂使用制作本作品的全部或部分的数字或硬拷贝, 但不收取任何费用, 前提是拷贝不是为了营利或商业利益而制作或分发, 并且拷贝带有本通知和首页上的完整引文。必须尊重 ACM 以外的其他人拥有的本作品组件的版权。允许以信用摘录, 要以其他方式复制或重新发布, 在服务器上发布或重新分发到列表, 需要事先获得特定许可和/或收费。从 permissions@acm.org 请求权限。

ESEC/FSE 21, 2021 年 8 月 23 日至 28 日, 希腊雅典
© 2021 计算机协会。
ACM ISBN 978-1-4503-8562-6/21/08. . . 15.00 美
元 <https://doi.org/10.1145/3468264.3468562>

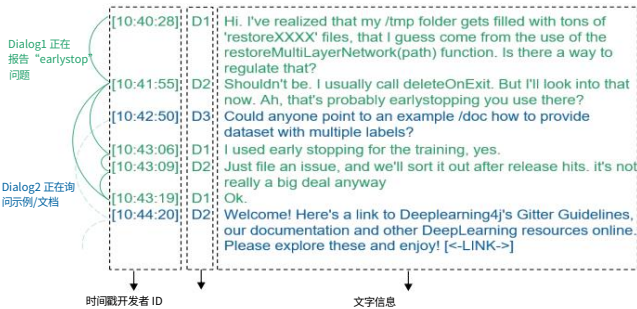


图 1: 来自 DL4J 社区的一段实时聊天日志。

ESEC/FSE 21, 2021 年 8 月 23 日至 28 日, 希腊雅典

林石, 萧晨、叶阳、姜寒之、姜子友、南牛、王青

如 OSS 开发者互动时,什么是社区结构,讨论哪些话题,以及 OSS 开发者如何互动,可以从海量的实时聊天数据中得出,这对于从高效的沟通方式中学习知识、改进现有的实时聊天平台非常重要,并指导促进高效和有效的 OSS 协作的研究方向。

尽管一些实证研究开始提倡使用对话来理解开发人员的行为[40,59,67],但很少关注开发人员在实时聊天中的沟通方式和内容。Shihab等人报告了最相关的研究。[58]在 Internet Relay Chat 上分析内容、参与者和通信方式。但是,他们的主题是 IRC 会议日志,这在许多方面与开发人员实时聊天对话不同。Parra等人的另一个相关工作线程。[49]和 Chatterjee 等人。[15]分别在 Gitter 和 Slack 中展示了两个开源开发人员通信数据集,目的是强调实时开发人员通信是未开发的信息资源。这促使我们的研究深入了解开源软件中开发人员通信的性质。

在本文中,我们对开发人员在 Gitter 上的实时聊天进行了首次全面的实证研究,调查了四个特征:他们交互的时间(通信配置文件)、社区结构是什么样的(社区结构)、讨论了哪些主题(讨论主题),以及它们如何交互(交互模式)。为此,我们首先收集了来自8个热门社区的大量开发者日常聊天记录。然后我们手动解开 749 个对话,并根据它们对 749 个对话的评估结果从四个最先进的模型中选择最佳解开模型。在自动解开之后,我们对实时聊天进行了实证研究,旨在揭示四个特征:通信概况、社区结构、对话主题和交互模式。我们总共研究了 173,278 条对话,1,402,894 条话语,来自八个开源社区的 95,416 名用户。

主要结果包括:(1)开发者在工作日比周末更有可能聊天,尤其是在周三和周四(UTC);(2)在OSS实时聊天社区中观察到三种社交模式:北极星网络、星座网络和银河网络;(3)开发者在实时聊天中经常讨论的前三个话题是API使用、错误和背景信息;(4)在线聊天中识别出六种交互模式,包括探索解决方案、澄清答案、澄清问题、直接/讨论答案、自我回答独白和未回答独白。本文的主要贡献如下。

- 我们对开发人员的实时聊天消息进行了第一次大规模分析研究,提供了基于经验的定量和定性结果,以更好地了解开发人员的沟通概况、社区结构、讨论主题和交互模式。· 我们为个人开发人员和 OSS 社区提供有关高效对话的实用见解,突出平台供应商所需的功能,并为研究人员阐明未来方向。· 我们提供实时聊天的大规模数据集¹,以促进我们的研究和未来应用的复制。

在本文的其余部分,第二部分说明了背景。第三部分介绍了研究设计。第四节描述了结果和分析。第五部分是对结果和有效性威胁的讨论。第六节介绍了相关工作。第七节总结了我们的工作。

2 背景

本节介绍相关的关键概念和技术。

2.1 Gitter 平台

许多 OSS 社区使用 Gitter [31]或 Slack [32]作为他们的实时通信手段。特别是,Gitter 是目前最受欢迎的在线交流平台[37],因为它提供对公共聊天室的开放访问和对历史数据的免费访问[49]。考虑到流行、开放和免费访问的性质,我们基于 Gitter2 进行了这项研究。Gitter 的社区通常有多个聊天室:一个普通聊天室和几个特定主题聊天室。通常,普通房间包含大多数参与者。在这项研究中,我们只关注普通房间。

Gitter 共有 2,171 个社区可以公开访问。截至 2020 年 11 月 20 日,2,171 个社区的参与者总数为 733,535 人。

本研究涉及关于 Gitter 实时聊天日志的三个主要概念,包括聊天日志、话语和对话日志。在 Gitter 中,开发人员在聊天室中的对话记录在聊天日志中。如图1所示,典型的实时聊天日志包含按时间顺序排列的一组连续话语。

每个话语都包含一个时间戳、开发人员 ID 和一个文本消息,用于发起问题或响应较早的消息。聊天日志通常包含大量话语,并且在任何给定时间,多个连续话语可能会响应对话讨论的不同线程。话语的交错性质导致对话纠缠,如图1中的两种颜色所示。这两种颜色用于突出属于两个不同对话的话语,其中两个话语之间的链接表示响应关系。

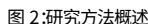
2.2 聊天分析中的挑战与许多其他与软件

开发相关的通信来源不同,在线通信平台上的信息以非结构化、非正式和交错的方式共享。因此,由于以下障碍,分析实时聊天非常具有挑战性。(1) 纠缠对话。聊天日志中的话语形成流信息,对话经常纠缠在一起,例如单个对话与其他对话交错,如图1所示。如果不将话语划分为一组不同的对话,就很难执行任何类型的高级对话分析对话框。

(2) 昂贵的人力。聊天日志通常是大量的,并且包含涵盖广泛的技术和复杂主题的非正式对话。分析这些对话需要经验丰富的分析师花费大量时间,以便他们能够彻底理解这些对话。因此,对开发人员的实时聊天进行全面研究是非常昂贵的。(3) 噪声数据。存在嘈杂的话语,例如重复和不可读的消息

¹<https://github.com/LiveChat2021/LiveChat#5-download>

² 在 Slack 中,社区由团队管理员控制,而在 Gitter 中,对聊天数据的访问是公开的。



社区	领域	全部的		样本			
		人口	UPDU	人口			
角[30]	前端 框架	22,467	79,619	695,183	125,977	78	
Appium[25]	移动的	3,979	4,906	29,039	73,877	724	
DL4J[27]	数据科学	8,310	27,256	252,846	93,100	1,130	
码头工人[21]	开发运维	8,810	3,954	22,367	74	90	1,126
以太坊[24]	区块链 平台	16,154	17,298	91,028	116,965	516	
吉特[42]	协作 平台	9,260	7,452	34,147	87,865	515	
节点[16]	Web应用程序 框架	18,118	13,981	81,771	144,987	737	
打字稿[17]	编程 语	8,318	18,812	196,513	110,951	5,170	
	全部的	95,416	173,278	1,402,894	822,749	7,226	

ESEC/FSE 21, 2021 年 8 月 23 日至 28 日, 希腊雅典

这些社区。Gitter 提供 REST API [28]来获取有关聊天室和发布话语的数据。在这项研究中,我们使用 REST API 来获取八个选定社区的聊天话语,检索到的数据集包含截至“2020-11-20”的所有话语。

详细统计如表1所示,其中P表示参与者的数量,D表示对话的数量,U表示话语的数量。八个社区的参与者总数为 95,416 人,占Gitter 总人口的 13%。因此,我们认为这八个社区是 Gitter 平台的代表。

预置文本话语。我们首先将非ASCII 字符(如表情符号)标准化为标准 ASCII 字符串。一些低频标记对实时聊天的分析贡献不大,例如URL、电子邮件地址、代码、HTML 标签和版本号。

我们将它们替换为特定的标记 <URL>、<EMAIL>、<HTML>、<CODE> 和 <ID>。我们利用 Spacy [2]将句子标记为术语,并使用 Spacy 对术语进行词形还原和小写,以减轻单词形态的影响。

对话解开的手动标签。我们采用 3 步手动过程来生成示例对话数据集以供进一步分析。首先,我们从每个社区的实时聊天日志中随机抽取 100 个话语,目的是跟踪与100 个话语中的每一个相关的相应对话。这一步从八个社区的全部 1,402,894 个话语中得到总共 800 个话语。接下来,使用每个话语作为种子,我们迭代地识别其前面和后续的话语,以便我们可以将相关的话语尽可能完整地分组到同一个对话中。具体来说,对于每个话语,我们通过检查聊天日志中的连续聊天来确定其上下文,并手动将其链接到属于同一对话的相关话语。然后,下一步是清洁。具体来说,我们排除了不可读的对话:(1)用非英语语言编写的对话;(2)包含过多代码或堆栈跟踪的对话框;(3)低质量的对话,例如错别字多、语法错误的对话;(4)

涉及频道机器人的对话。在这些步骤之后,我们包括与最初的 800 条话语相关联的另外 6000 条话语。这导致总共 7,226 条话语,人工分解成 749 个对话,如表 1 所示。请注意,删除涉及机器人的对话对我们的结果几乎没有影响。

首先,涉及机器人的对话量相对较小。在我们的研究中,八个项目中只有一个使用了机器人,更具体地说,由于机器人的参与,800 个抽样对话中只有九个被排除在外。其次,我们观察到机器人生成的话语相当琐碎,例如问候信息、一般指南的链接和状态更新。

为确保解开结果的正确性,组建了一支由一名高级研究员和六名博士组成的拉贝尔团队。学生。他们都精通英语,并且在软件开发方面进行了深入的研究工作,或者一直在积极为开源项目做出贡献。高级研究员培养了六名博士。候选人如何解开对话并在此过程中提供咨询。解开纠缠的结果来自博士学位。其他人对候选人进行了审查。当对话完全同意时,我们才接受对话并将其包含到我们的数据集中。当对话收到不同的解缠结果时,我们与所有团队进行了讨论

林石, 董晨、叶阳、姜寒之、姜子友、南牛、王青

成员通过投票决定。关于对话解开的平均 Cohen 的 Kappa为 0.81。

自动对话解开。为了大规模分析对话,我们尝试了 2.2 节中介绍的四种最先进的 DD 方法(即 BiLSTM 模型、Bert模型、E2E 模型和 FF 模型)。具体来说,我们使用上一步的手动解缠结样本数据作为ground truth数据,比较并选择最佳的DD模型,以便在本研究中进一步分析。我们实验的比较结果表明,FF 方法通过在所有指标上获得最高分,在解开开发人员实时聊天方面明显优于其他方法。NMI、Shen-F、F1 和 ARI 的平均得分分别为 0.74、0.81、0.47 和 0.573。最后,我们使用最好的 FF 模型来解开聊天日志中的所有 1,402,894 个话语。我们总共获得了 173,278 个对话。

3.3 实证分析设计3.3.1 RQ1 (通信配置文件)

分析。由于良好的沟通习惯暗示了更有成效的开发实践,我们打算揭示开发人员的时间沟通概况,包括开发人员何时活跃以及受访者对对话发起者的回复时间。首先,我们收集了整个人群的所有话语时间,并分析了高峰时段和高峰天数。然后我们计算响应时间延迟:

$$T_{\text{response}} = t_{\text{reply}} - t_{\text{init}} \tag{1}$$

其中是发起者发起对话的时间,是第一响应者回复的时间。我们自动计算所有 173,278 个对话的话语时间和响应时间滞后。

3.3.2 RQ2 (社区结构)分析。我们的目标是在实时聊天中可视化开发人员的社交网络,并总结常见的结构。社交网络分析(SNA)描述了社会实体之间的关系,以及它们之间联系的结构和含义[65]。为了研究一个 OSS 社区中开发人员之间的关系,我们根据以下定义生成社交网络:

$$\begin{aligned} &= \{, \} = \\ &\{1, 2, \dots, \} = \{ <, \\ &> \} \end{aligned} \tag{2}$$

where 是聊天室中的开发者,是对话发起者,是。具体来说,对于每个解开的对话,我们首先确定其发起者和所有受访者。发起者是发起对话的开发者,响应者是参与对话的其他开发者。然后我们在发起者和每个响应者之间添加一个链接。(注意,RQ2 侧重于探索响应行为,即发起者和响应者之间的交互,因此仅在这两个角色之间定义了链接/边。此外,我们将在RQ4中探索发起者和所有响应者之间的交互关系,关注讨论行为。)最后,我们在构建社交网络时使用了一个未加权的图,用于可视化所有开发人员在实时聊天中的关系。社交网络可以表现出网络的连通性和密度

3由于篇幅关系, Github上提供了评估现有DD模型的实验细节 :https://tinyurl.com/3dyu5n44

开源社区。我们使用自动图形工具 Gephi [6]基于所有 173,278 个对话构建社交网络。

为了解八个社交网络的拓扑结构,我们报告了以下已被先前研究广泛使用的 SNA 度量[45, 55]。请注意,我们在根据之前的工作[11, 62]计算 SNA 测量时排除了从未收到回复 (AKA. Haircut)的开发人员。(1) Degree [20]测量边缘的数量。(2)介数[26]衡量一个开发人员位于其他开发人员之间最短路径上的频率。

(3) Closeness [7]衡量一个开发人员与所有其他开发人员的平均距离 (反距离)。(4)直径[13]是连接网络中最大的测地线距离。(5)聚类系数[66]是衡量图中开发人员倾向于聚集在一起的程度。

3.3.3 RQ3分析 (讨论主题)。我们的目标是确定开发人员实时聊天中的讨论主题。查特吉等人。[15]观察到实时聊天提供的信息与 Stack Overflow 上的问答帖子中的信息相似。为了有效地识别和组织对话主题,我们在 Stack Overflow [9]上扩展了 Beyer 等人的问题类别。我们选择 Beyer 等人的类别有两个原因。首先,由于开发人员同时使用问答论坛 (如 Stack Overflow)和实时聊天来解决开发问题,我们认为 Stack Overflow 问题的类别部分适用于实时聊天中的对话主题。其次,Beyer 等人的类别协调了先前研究[3,8,10,51, 64]中提出的五种分类法,这些分类法已经过验证并适用于开发人员问题的帖子。

尽管如此,预定义类别并不意味着是全面的,因此,我们采用混合卡片分类过程[22]来手动确定对话的主题。在混合卡片分类中,分类从预定义的 Beyer 等人的类别开始,参与者也可以创建自己的类别。新创建的主题会立即更新到主题集中,供其他参与者使用。参与者是手动解开对话的同一个团队,标记过程类似于第 3.2 节中介绍的手动解开对话。具体来说,排序过程在一轮中进行,并有一个结束讨论环节,以基于多数投票解决标签上的分歧。关于对话主题的平均 Cohen 的 Kappa 为 0.86。

3.3.4 RQ4 (交互模式)分析。实时聊天对话通常用于解决方案探索和讨论刺激的目的。为了揭示塑造和/或指导更高效对话的潜在模式,我们首先采用开发者意图代码本[50]并手动标记每个对话中出现的交互连接。开发者意图代码本

表 2:实时聊天中的开发者意图类别

代码	标签	描述
OQ 原始问题	开发者发起对话的第一个问题	
CQ 澄清问题	开发人员要求澄清	
FD 更多详情	开发者提供更多细节	
FQ 跟进问题	开发者就相关问题提出跟进问题	
PA 潜在答案	开发人员提供的潜在答案或解决方案	
PF 正反馈	开发人员为工作解决方案提供积极的反馈	
NF Negative Feedback Developer	为无用的解决方案提供负面反馈	
GG 问候/感谢	问候或表示感谢	

是基于先前关于信息搜索对话中用户意图的工作[50] 构建的,如表2 所示。

然后,我们采用开放式卡片排序[52]过程,根据开发人员的意图顺序将交互模式分配给对话。在开放式排序中,排序从没有预定义的模式开始,参与者开发自己的模式。两个参与者分别将模式分配给相同的对话。分拣过程分多轮进行。在第一轮中,所有参与者标记一个社区的对话,并进行密集的讨论以实现关于模式的概念一致性。

使用和仔细维护共享模式池,每个参与者都可以从共享池中选择现有模式和/或将新模式名称添加到共享池中。然后我们分成两个团队来标记剩余的对话。每个对话都会收到两个模式标签,我们会根据多数投票解决分歧。关于交互模式的平均 Cohen 的 Kappa 为 0.82。

在确定了潜在的交互模式之后,我们进一步探索了它们在分布和持续时间方面的统计特征。我们计算对话的持续时间如下:

= - (3)

where 是对话结束的时间,是发起者启动对话的时间。该指标可以反映一个对话的生命周期。

请注意,为了使手动标记每个对话的工作量保持可控,我们通过手动分析749 个采样对话来回答 RQ3 和 RQ4。我们相信,尽管我们只能手动分析一小部分解开的对话,但该数据集支持我们的方法,因为它有助于发现有价值的发现。

4 结果与分析

4.1 RQ1:沟通概况为了回答这个问题,我们分析了两个指标,即话语时间和响应时间。接下来,我们报告了在八个 Gitter 社区中比较这些指标的结果。

话语时间。图3(a)比较了 8 个社区 24 小时内话语强度的分布。

首先,我们用红色虚线圆圈标识每个社区的高峰时段,然后用黄色阴影突出显示基于其中包含的高峰时段的时间窗口。我们可以看到,高峰时段有 3 个窗口,分别是 UTC 9 点到 10 点、13 点到 14 点、18 点到 21 点。另外,UTC 1 到 6 点对应的是聊天活动的低谷时段。开发者在那个时候聊天不太活跃。

图3(b)显示了话语在不同工作日的分布。我们可以看到开发人员在工作日比周末 (UTC)聊天更多。

响应时间。图3(c)展示了根据 8 个社区的 173,278 次对话计算得出的响应时间分布。平均响应时间为 220 秒,最大时滞为1,264 秒,最小时滞为 2 秒。峰值为 (23, 393),即 23 秒内有 393 个对话得到回复。我们可以看到,时滞从0 到 23 秒大幅增加,并呈长尾下降。百分之八十的对话在 343 秒内得到第一个响应。正如最近一项关于 Stack Overflow [43]的研究报告的那样,快速回答的阈值是439 秒。相比之下,实时聊天的速度提高了 50% ((439-220)/439)

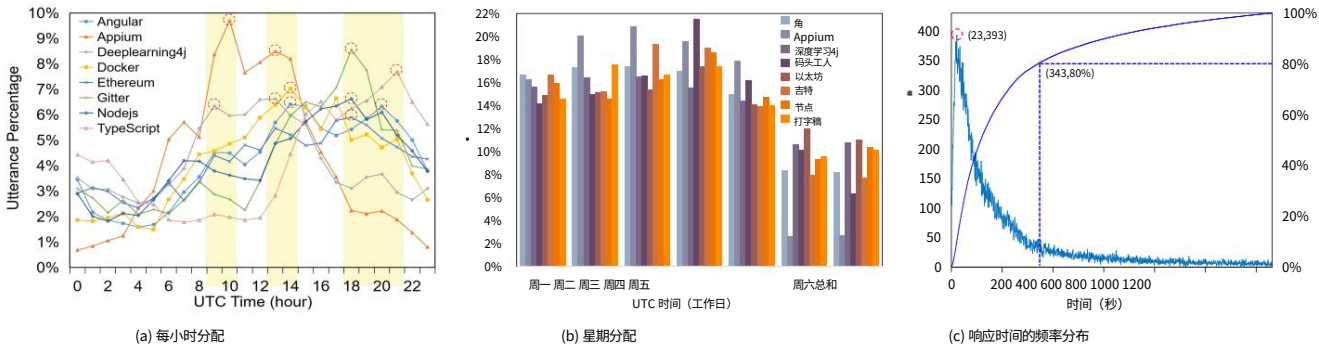


图 3:关于通信配置文件的统计结果

回复比堆栈溢出中的快速答案。因此,我们认为来自实时聊天的响应相对较快。

回答 RQ1:实时聊天的高峰时间是UTC 9到10、13到14、18到21,而UTC 1到6是低活跃时间。
与周末相比,开发人员更有可能在工作日聊天,尤其是在周三和周四 (UTC)。此外,实时聊天的回复速度比 Stack Overflow 中的快速回复快 50%。

4.2 RQ2:社区结构为了回答RQ2,我们首先检查了

八个社区的开发者社交网络的结构特性,然后尝试根据这些社交网络得出一些共同的观察结果。

社交网络的属性。表3显示了八个社区的社交网络属性。Init.%,Resp.%和 Both%表示担任对话发起者、受访者和两者角色的开发人员的百分比。直觉上,我们认为受访者与他人分享他们的知识,而发起者则从他人那里获得知识。我们可以看到,四个社区 (Appium、Docker、Gitter 和 Ethereum)的对话发起者年龄百分比比较高 (75.04%-81.70%),受访者百分比比较低 (18.30%-24.96%) /两者。对话发起人比例高可能与开源项目的适用性有关,例如以太坊是使用最广泛的开源区块链系统之一,因此有大量用户从实时聊天中获得技术支持。其他四个社区 (Angular、DL4J、Nodejs 和 Typescript)的受访者比例更高 (29.94%-48.62%) /两者都有。一种可能的解释是,这四个项目更广泛地用于开发目的,例如,Angular 是一个用于构建移动和桌面 Web 应用程序的平台,因此,这些社区似乎是知识共享和协作的。

对开发者社交网络进行分类。图4显示了Gephi生成的八个社区的社交网络可视化。每个节点代表一个开发者,边de记录了两个开发者之间的对话关系。我们将发起者的顶点用蓝色着色,将响应者的顶点用白色着色,并将两个角色的顶点用橙色着色。此外,节点的大小表示其对应的程度。基于对社区结构的观察,我们将八个社区分为三组,包括: (1) Polaris 网络是一种高度中心化的网络,社区围绕其单一焦点进行组织或组织; (2)星座网络是一种适度中心化的网络,其中社区

围绕其多个焦点进行组织; (3)银河网络是一种去中心化网络,社区中的所有个体都具有相似的关系。在图4 中,顶部的四个社区 (即 Angular、DL4J、NodeJS 和 Typescript)属于 Constellation 网络,即适度集中的网络。三个社区 (即 Appium、Docker 和 Gitter)对 Polaris 网络,即高度中心化的网络很感兴趣。剩下的以太坊社区属于银河网络,即去中心化网络。先前的研究表明,高度集中的网络可能反映了整个社区的知识分布不均,其中知识主要集中在焦点 [38, 44]。因此,如果焦点开发者不活跃,三个 Polaris社区 (Appium、Docker 和 Gitter)可能存在较高的单点故障风险,而 Galaxy 网络 (以太坊)的风险最低,其次是 Constellation 网络 (Angular、DL4J、NodeJS 和 Typescript)。

在表3 中,我们还可以看到 Constellation 网络和Polaris 网络在平均度数 (1.96-9.15)、介数 (0.000273-0.001342)和接近度 (0.31-0.43)方面得分更高。这些现象表明星座网络和北极星网络中的焦点使社区更加复杂。

连接。一项关于电子邮件连接社交网络的研究[11]表明,开发人员的平均介数为 0.0114,平均高于实时聊天社区。具有高介数的节点在允许信息从网络的一部分传递到另一部分方面可能在网络中具有相当大的影响。较低的介数表明实时聊天中的开发人员在传播信息方面的影响可能小于电子邮件中的开发人员。

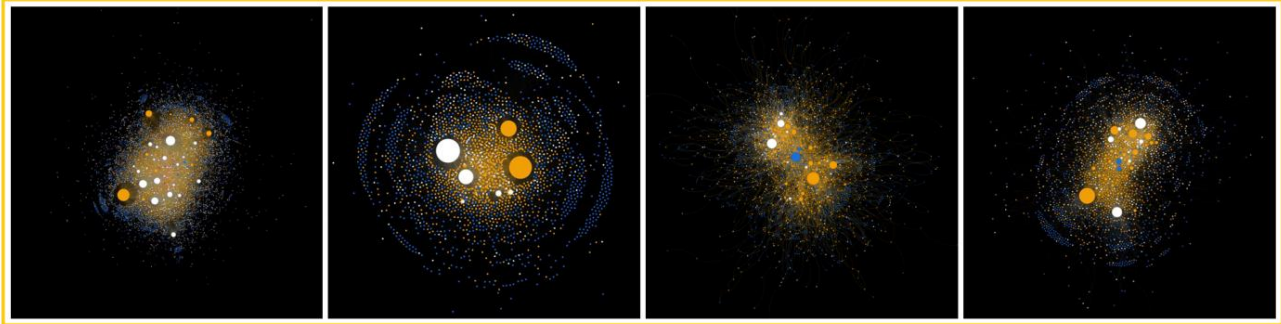
但是,基于电子邮件构建的网络的平均入度和出度明显较低,开发人员分别为 0.00794 和 0.00666。虽然实时聊天中的开发人员具有更高的集中度和更高的密度,以及接近中心性值,这表明社区比电子邮件中的社区联系更紧密。

Constellation 社区的开发者聚类系数得分较高 (0.14-0.60),说明 Constellation 社区的开发者联系更紧密,即Angular、DL4J、Nodejs 和 Typescript 的开发者之间的相互了解程度更高。

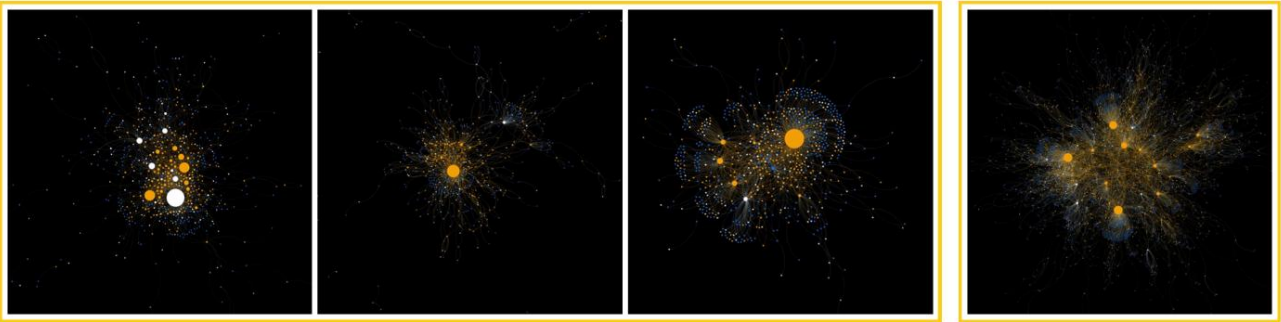
回答 RQ2:通过可视化所研究的八个社区的社交网络,我们确定了三个用于开发人员实时聊天的社交网络结构。一半的社区 (4/8)是星座网络。少数社区 (3/8) 是 Polaris网络。只有一个社区属于 Galaxy 网络。在

表 3:八个社区的社交网络测量

	星座				北极星			星系
	Angular DL4J	Nodejs Typescript	Appium	Docker Gitter 以太坊				
在星图: %	54.54%	51.38%	70.06%	56.94%	75.04%	76.22%	75.33%	81.70%
响应: %	13.90%	11.39%	6.83%	12.42%	7.43%	5.84%	9.04%	5.59%
两者 %	31.56%	37.23%	23.11%	30.65%	17.53%	17.94%	25.62%	12.71%
度	9.15	5.02	2.75	4.2	2.59	1.96	2.07	1.92
介数	0.000273	0.000867	0.000468		0.000786	0.001035	0.001173	0.001342
亲近	0.35	0.42	0.31	0.34	0.35	0.37	0.43	0.34
直径	8	6	13	9	10	10	8	15
聚类系数	0.41	0.6	0.14	0.33	0.2	0.13	0.2	0.16



(a) Constellation {从左到右:Angular,DL4J,Nodejs,Typescript}



(b) Polaris {从左到右:Appium,Docker,Gitter}

(c) 银河 {以太坊}

图 4:八个开发者实时聊天社交网络的可视化

对比一下,我们发现实时聊天中的开发者可能会少一些比电子邮件中的开发人员在传播信息方面的影响力更大,但是拥有比电子邮件更紧密的社区。

4.3 RQ3:讨论主题

图5开发者直播中讨论话题的分布聊天。该图以灰色显示讨论主题及其类别白色,以及相应对话框的百分比。分类法从更高级别的类别向外扩展至较低级别的类别和主题。在这项研究中,我们扩展了 Beyer 等人的类别,以适应开放和实时聊天: (1) 添加社交聊天和一般开发类别,和 (2) 将“概念”和“差异”类别分解为区分更有价值的信息,例如不需要的行为和新功能。有关对话主题的更多信息,我们提供包含详细信息和示例的公共 Github 存储库4。

4<https://github.com/LiveChat2021/LiveChat#34-rq3-discussion-topic>

最内圈表明,在所有八个社区中, 89.05% 的对话是与领域相关 (DR) 的主题,例如主题与社区的业务领域相关,而 10.95% 的对话是非领域相关 (N-DR) 主题,例如一般开发或社交聊天。 DR主题可以进一步分解根据用途不同分为三类。这些包括以解决方案为导向的对话,占比最高 (35.25%) ,其次是以问题为导向的对话 (32.98%)和面向知识的对话 (20.83%)。在 35.25%的面向解决方案的对话中, 29.37% 是关于 API 使用的,5.87% 是关于审查。在 32.98% 的问题导向对话中,大部分 (20.29%) 讨论差异,包括不受欢迎的行为,不工作、可靠性问题、性能问题和测试/构建失败。我们可以看到,开发人员讨论了更多的“不良行为”和“做不工作”,而不是可靠性问题、性能问题和测试/构建失败。在 20.83% 的知识型对话中,大部分他们 (13.75%)讨论概念,包括背景信息,

ESEC/FSE 21, 2021 年 8 月 23 日至 28 日, 希腊雅典

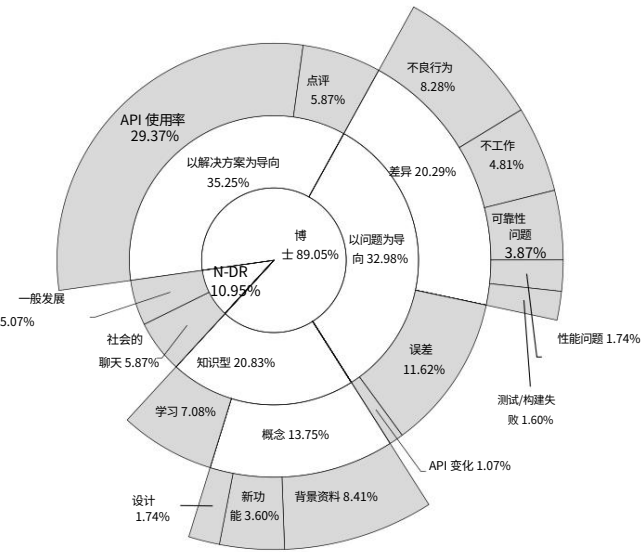


图 5: 从中心向外阅读的开发者实时聊天中的讨论主题分布

新功能和设计。总体而言,前三个常见主题是API 使用 (29.37%)、错误 (11.62%) 和背景信息 (8.41%)。

回答 RQ3: 开发人员推出面向解决方案的对话和面向问题的对话,而不是面向知识的对话。将近 1/3 的对话是关于 API 使用的。与可靠性问题、性能问题和测试/构建失败相比,开发人员讨论的错误、不良行为和不工作更多。

4.4 RQ4: 交互模式交互模式。图6展示了

实时聊天中的六种交互模式,使用第 3.3.4 节中介绍的开放卡片排序构建。此图在蓝色节点中显示对话发起者,在黄色节点中显示响应者。线条表示回复关系,标签表示表2 中的开发人员意图。在这项工作中,我们确定了以下六种交互模式: (1)

P1: 探索解决方案。鉴于对话发起者发布的原始问题,其他开发人员提供了可能的答案。但

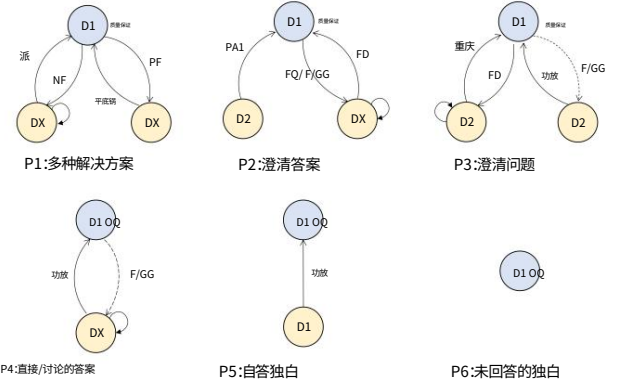


图 6: 交互模式, F 表示反馈, 包括负反馈和正反馈, 虚线表示可选交互。

林石, 萧晨, 叶阳, 姜寒之, 姜子友, 南牛, 王青

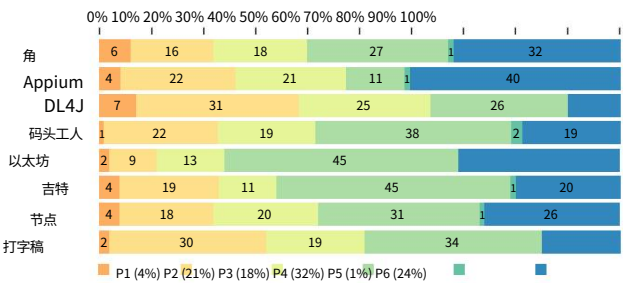


图 7: 不同社区之间的互动模式分布

发起人给出负面反馈,表明这些答案没有解决问题。当发布正确答案时,发起者给出积极的反馈并结束对话。(2) P2: 清晰的答案。鉴于对话发起者发布的原始问题,另一位开发人员提供了可能的答案。然后发起人发布后续问题以澄清答案,直到发起人完全理解为止。(3) P3: 澄清问题。鉴于对话发起人提出的原始问题,被访者要求发起人更详细地澄清问题,直到他们完全理解为止。然后受访者发布答案,发起者给出反馈或问候。(4) P4: 直接/讨论答案。给定对话发起者提出的原始问题,被访者直接回答,或在内部讨论后回答。(5) P5: 自答独白。对话发起人发布的原始问题由他或她自己回答。

(6) P6: 未回答的独白。对话发起人发布的原始问题没有得到回答。

模式的百分比。图7显示了不同社区中交互模式的百分比,图例中显示了平均百分比。P1~P6 指的是上面定义的六种交互模式。我们可以看到,直接/讨论答案 (P4) 模式在大多数社区中所占比例最大。

此外,我们注意到相当多的对话 (1%) 属于自答独白,而 24% 的对话属于未答独白。近 1/4 的对话在实时聊天中没有得到回应。

我们将在 5.1 节讨论更多的独白。

模式的持续时间。图8显示了小提琴图以及每种模式的持续时间分布。P1~P5 指的是上面定义的交互模式。这里我们只展示五种模式,因为

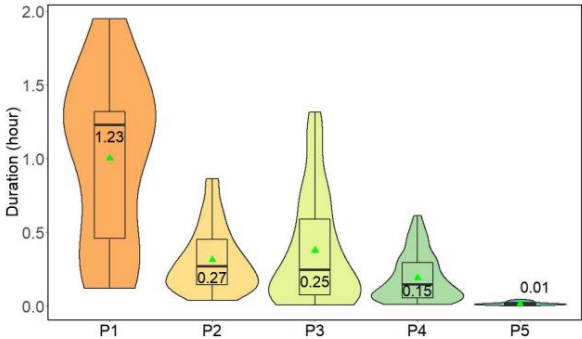


图 8: 交互模式的持续时间分布,除了未回答的独白。

P6 指的是几乎没有持续时间的未回答的独白。我们可以看到, 虽然 P1 在对话中所占的比例很小, 但持续时间最长。其平均持续时间为 1.00 小时。P2 和 P3 的持续时间略长于 P4。P5 持续时间最短, 平均持续时间为 0.02 小时。

回答 RQ4: 在实时聊天中确定了六种交互模式: 探索解决方案、澄清答案、澄清问题、直接/讨论的答案、自我回答的独白和未回答的独白。直接/讨论答案模式在大多数社区中所占比例最大。平均仍有 1/4 的对话没有得到响应。属于探索解决方案模式的对话比其他对话持续时间最长。

5 讨论

在这项工作中, 我们首先从沟通概况、社区结构、讨论主题和交互模式等方面来了解开发人员在 Gitter 上的实时聊天。我们的工作为其他研究人员能够在其他研究中使用相同的方法铺平了道路

软件社区。此外, 由于沟通是成功软件开发的重要组成部分, 而 Gitter 是 GitHub 用户沟通的主要平台之一, 因此探索软件工程师如何使用 Gitter 以及他们使用它的痛点非常重要。为了促进高效和有效的 OSS 通信, 我们讨论了我们的发现对 OSS 开发人员、社区、平台供应商和研究人员的主要影响。

5.1 个人开发者根据我们的研究结果, 我

们提出了以下对个人 OSS 开发者有效和高效地吸引注意力和接收响应的启示。

(1) 在寻求解决方案帮助时提供示例代码或数据 (RQ3、RQ4)。在图 5 中, 我们报告说, 有近 1/3 的对话是面向问题的。在实时聊天中, 重要的是在面向问题的对话框中提供示例代码, 以使其他开发人员快速理解并避免丢失关键信息。

这一发现也与之之前关于 Stack Overflow 的研究 [12, 14] 提供的证据一致。这是一个显示的例子

<D1> 嘿, 任何人都对这个问题有想法: 在 UI 中显示引导。我试图通过在 .ts 文件中替换它来删除它..... 我不确定还有什么可以尝试的。

<D2> 我无法理解发生了什么。你能把你的例子扔在一个笨拙的地方吗?

在实时聊天中使用示例的重要性。相应的对话通过多次来回交互来确认澄清问题模式 (P3), 以澄清问题。如果在开始时提供示例, 将加快解决问题的过程。特别是, Angular 社区非常重视示例演示。鼓励开发人员通过名为 Plunker [1] 的演示平台创建示例。

(2) 注意低活跃时间 (RQ1)。我们的结果表明, 开发人员在实时聊天的某些时间段内更加活跃。图 3(a) 显示最活跃的时间片是 UTC 9-10、13-14 和 18-21, 对应于中欧/美洲白天或亚洲夜间。值得注意的是, 与其他工作日 (UTC) 相比, 周三和周四发生的开发者实时聊天更多, 这可能对应于沟通、协调、

并为周五的集成/发布截止日期做准备。这一观察结果也证实了最近研究中报道的 OSS 项目的“商业上可行的替代方案” [23, 29, 68]。常见的发现之一是, 由自愿开发人员推动的 OSS 项目的传统概念现在已经过时。OSS 已成为一种商业上可行的替代方案, 一些 OSS 项目已成为全球组织的关键组成部分。

例如, Docker 被世界各地的软件公司广泛使用, 包括 Adobe、AT&T、PayPal 等。因此, 开发问题可能不会在周末聊天, 而是在工作日在实时聊天中讨论他们

而低活跃时间片 (UTC 1-6) 主要对应于中欧/美洲夜间或亚洲白天时间。如果开发人员在低活跃时间发现问题并需要支持, 我们建议几个选项。首先, 建议将问题同时发布到其他替代平台, 例如问题和电子邮件。其次, 如果没有及时收到对在低活跃时间发布的问题的回复, 他们最好在实时聊天中跟进。最后, 使用一些自动提醒机器人, 例如, 查看在低活跃时间发布的问题列表。

(3) 避免在正在进行的讨论中提问 (RQ4)。在从对话中识别出未回答的独白模式时, 我们注意到其中 30% 的独白模式是在针对不同主题的持续活跃和激烈对话中启动的。在这种情况下, 新的问题很容易被正在进行的讨论所淹没。因此, 为了增加获得回复的机会, 开发人员可以在持续讨论之后发布他们的问题。

如果出现紧急问题, 我们建议平台供应商提供特殊的便利来标记这种紧急情况并将团队的注意力转移到它上, 例如多线程对话 (例如, 在 Slack 中) 或紧急问题的突出显示标签和让其他人监督紧急标签的使用, 以避免滥用。

5.2 OSS 社区

我们为 OSS 管理者提供以下建议, 以改善社区的管理和协调。

(1) 降低单点故障的风险 (RQ2)。正如 RQ2 中所报告的, 如果重点开发人员离开或变得不活跃, 三个 Polaris 社区 (Appium、Docker 和 Gitter) 可能会面临更高的单点故障风险。值得注意的是, 在三个 Polaris 网络中, 有一些第二个焦点小于那些最重要的焦点, 这可能会提出一些实用的策略来降低单点故障的风险。例如, Polaris 社区可能会设计和采用适当的激励措施或政策来支持第二重点开发人员, 以提高实时聊天社区的弹性。

(2) 为新手改进 OSS 文档 (RQ1、RQ4)。据报道, 一些新人抱怨很难开始一个新项目并获得其他社区成员的及时帮助, 这可能会使他们逐渐失去动力, 甚至放弃贡献 [63]。为了方便新手熟悉自己并以更有效的方式做出贡献, OSS 社区可以考虑利用我们的研究结果来改进 OSS 文档。例如, RQ1 的结果显示了活动和低活动时间片, 而 RQ4 的结果显示了许多

ESEC/FSE 21,2021 年 8 月 23 日至 28 日,希腊雅典

林石, 萧晨、叶阳、姜寒之、姜子友、南牛、王青

在正在进行的讨论中,无人回答的独白正在问。这些信息可以被纳入新人的自述文件中,他们希望为项目做出贡献,以及如何从他人那里获得及时的帮助。

5.3 平台供应商

本节讨论了促进更富有成效的对话的几个所需功能。具体来说,这些是从通信平台供应商的角度组织的,以支持更智能和高效的聊天选项,利用密集历史对话的挖掘和知识共享。

(1) 突出和组织对话主题 (RQ3)。正如之前的研究所建议的,关注点的多维分离[47]是一个强大的概念,通过将大型讨论分解为许多较小的单元来支持协作开发。它强调,在线交流平台供应商可以为一组专注于特定主题的预定义面板提供支持。在 RQ3 的结果中,我们提供了实时聊天中讨论主题的分类。在线交流平台供应商可以参考此分类来创建主题面板。例如,API 使用面板、错误面板、背景信息面板等。

这些多个小组可以为社区成员带来以下好处:(i)快速了解和检索对话发起者的意图;(ii)减少干扰,更多地关注感兴趣的话题;(iii)识别开发商报告的重要信息。

(2) 注释重要问题 (RQ3)。在研究对话主题时,我们注意到某些类型的对话为未来的软件发展提供了信息。例如,有 3.6% 的对话讨论新功能,8.28% 讨论不受欢迎的行为。这些对话对于产品团队规划未来版本很有价值。同时,其他类型的对话表明现有系统的未揭示缺陷。例如,11.62% 的对话讨论错误,4.81% 讨论不起作用的东西,3.87% 讨论可靠性问题,1.74% 讨论性能问题。用“功能请求”、“增强”和“错误”正确注释这些对话框将有助于保留有价值的信息,并有助于提高软件的生产力和质量。例如,已经探索了挖掘实时聊天的技术,以从聊天日志中识别特征请求[57]。

5.4 研究人员

SE 领域的研究可以致力于在以下方向促进高效和有效的 OSS 通信。

(1) 自动推荐类似问题 (RQ4)。现有的在线交流平台只记录海量的历史聊天消息,并没有考虑对这些历史数据进行更深层次的利用。实际上,我们注意到开发人员有时会在实时聊天中发布类似的问题。在 DL4J 中,一位发起人发布了一个问题,他得到了这样的回复:“前段时间有人问了一个非常相似的问题。”然而,发起者要从海量的历史信息中准确地检索出类似的问题并不容易。此外,一些没有得到回答的问题很大程度上是由于之前回答了许多类似的问题。

因此,我们认为,如果研究人员能够开发出自动推荐类似问题和相应讨论的方法,将节省开发人员的工作量。

(2) 自动分配适当的受访者 (RQ4)。

通过分析属于探索解决方案模式 (P1) 的对话,我们注意到,对与发布的问题相关的技术不太熟悉的受访者可能会给出无效的解决方案。虽然这样的讨论可以让开发人员更好地理解问题,但多次失败和尝试的交互仍然延长了解决问题的过程。因此,为了使对话更有成效,期望开发出可以根据他们的历史答案推荐或分配适当受访者的方法。

(3) 自动将有价值的信息推送到项目存储库 (RQ3)。有价值的信息,例如功能请求或问题报告,无论是由开发人员手动注释还是由工具自动检测,都需要在项目存储库范围内进行详细记录和跟踪。通常,诸如 Github 或 Gitlab 之类的代码存储库提供了跟踪跟踪的功能。如果研究人员能够提供一种方便的方式将有价值的信息直接推送或集成到代码存储库中,效率会更高。此外,以下语言模式可能有助于自动分类对话主题。

据观察,API 使用类别中的问题包括诸如:“如何做某事?”、“我怎么做某事?”、“有人可以帮我做某事吗?”、“有什么办法吗?”或“我想要/需要做某事”。例如,“如何将我的 Angular 2 应用程序捆绑到 bundle.js 文件中?”和“任何人都可以使用 angular 2 帮助我处理 PWA”。来自错误类别的问题可能是“我收到/收到此错误”、“有人遇到过这样的错误”、“有人知道某事的解决方案吗”,或者直接发布特定的异常名称。

例如,“我收到 NotFound 错误”和“之前尝试加载路线时有人遇到过这样的错误吗?”背景信息类别中的问题可能是“什么/为什么/何时……”、“我想知道某事”或“是否有某事……”。例如,“Appium 什么时候支持 Xcode 8.2?”和“自动化使用 Swift 3 制作的 iOS 应用程序有什么限制吗?”。

(4) 分析社交聊天的效果 (RQ3)。正如 RQ3 中所报告的,10.95% 的对话是非领域相关 (N-DR) 主题,例如一般开发或社交聊天。最近的一项研究[46]强调了社交互动的重要作用,例如简单的短语“你周末过得怎么样?”,以显示在 COVID-19 大流行期间在家工作的开发人员的同伴支持。未来的工作可能会探索更多社交聊天的模式和效果,例如,大流行前后的比较。

5.5 对有效性的威胁外部有效性。外

部威胁与所提出方法的普遍性有关。我们的实证研究使用了来自 Gitter 的 8 个参与度最高的开源社区。尽管我们普遍认为所有社区都可能受益于从更有效、更有效的沟通方式中学到的知识,但未来的研究需要关注不太活跃的和

对所有类型的社区进行比较。

内部效度。内部威胁与实验错误和偏差有关。第一个威胁与我们采用的对话解开模型的准确性有关。尽管我们从最先进的方法中选择了最好的模型来解开对话,但最佳模型的准确度得分仍然不太令人满意。它将对 RQ1 和 RQ2 的结果产生影响。

为了解决这个问题,我们正在进行的一项工作是建立一个基于深度学习的新的对话解缠结模型,以提高现有解缠结方法的准确性。第二个威胁与随机抽样过程有关。抽样可能会导致不完整的结果,例如主题分类和交互模式。未来,我们计划扩大分析的数据集,并检查是否出现了新的主题或交互模式。

第三个威胁可能来自于人工拆解和卡片分类的过程。我们知道这样的过程可能会引入错误。为了减少这种威胁,我们建立了一个标签团队,并对每个结果进行同行评审。我们只采用获得完全同意的数据,或就不同的选项达成一致。

构造效度。构造威胁与评估指标的适用性有关。在这项研究中,手动标记主题和交互模式是一种构建威胁。为了最大限度地减少这种威胁,我们使用以前工作[5, 10, 53]使用的一种众所周知的方法来为文本软件工件构建合理的分类法。

6 相关工作

我们的工作与之前关注OSS 社区中同步和异步通信的研究有关。

OSS 社区中的同步通信。最近,越来越多的工作已经意识到通过现代 com 进行实时聊天

沟通平台在团队沟通中发挥着越来越重要的作用。林等人。[40]通过调查 104 名开发人员,就了解 Slack 在支持软件工程中的作用进行了探索性研究。他们的研究表明,开发人员将 Slack 用于个人、团队和社区范围的目的,开发人员在日常生活中使用机器人进行团队和任务管理。他们强调,实时聊天在软件开发中发挥着越来越重要的作用,在某些情况下取代了电子邮件。希哈布等人。[58, 59]从几个维度分析了两个大型开源项目的开发者 IRC 会议频道的使用情况:会议内容、会议参与者、他们的贡献和会议风格。他们的结果表明,IRC 会议在开源开发人员中越来越受欢迎,并强调了可以从开发人员聊天消息中获得的丰富信息。于等人。[67]分析了全球软件开发项目中两种通信机制的使用,即同步 (IRC)和异步 (邮件列表)。他们的结果表明,开发人员以互补的方式积极使用这两种通信机制。综上所述,现有的实时聊天实证分析主要集中在使用目的[40]、实时会议的使用[58,59],以及与不同通信机制和知识共享平台的比较[67]。缺乏对社区属性的深入分析和详细的讨论内容。我们的研究通过对实时聊天中的通信配置文件、社区结构、对话主题和交互模式的大规模分析来弥补这一差距。

OSS 社区中的异步通信。之前的研究对 OSS 社区中的异步通信进行了实证分析,包括邮件列表、问题讨论和Stack Overflow。伯德等人。[11]在Apache HTTP 服务器项目上挖掘电子邮件社交网络。他们报告说,电子邮件社交网络是一个典型的电子社区:少数成员占据了发送的大部分消息和大部分回复。迪

索博等人。[60]提出了一种意图分类法,将开发人员邮件列表中的句子分为六类:功能请求、意见询问、问题发现、解决方案建议、信息寻求和信息提供。尽管该分类已被证明在分析开发电子邮件和来自应用评论的用户反馈方面是有效的[48],但Huang 等人。[36]发现它不能推广到问题跟踪系统中的讨论,他们通过提出基于卷积神经网络的方法解决了 Di Sorbo 等人分类法的缺陷。艾莉亚等人。[4]通过对 Github 中 15 个问题讨论线程的定量内容分析,确定了 16 种信息类型,例如新问题和请求、解决方案使用情况等。他们还通过使用具有 14 个会话特征的随机森林对句子进行分类,提供了一个有监督的分类解决方案。Allamanis 和 Sutton [3]提出了一个主题建模分析,它结合了来自 Stack Overflow 的问题概念、类型和代码,以将编程概念和标识符与特定类型的问题相关联,例如“如何执行编码”。同样,Rosen 和 Shihab [51]使用基于潜在 Dirichlet 分配的主题模型来帮助我们总结Stack Overflow 中与移动相关的问题。我们的工作与现有研究的不同之处在于,我们专注于同步通信,这会带来不同的挑战,因为实时聊天日志是非正式的、非结构化的、嘈杂的和交错的。

7 结论和未来工作

在本文中,我们提出了第一个大规模研究,以获得对 OSS 开发人员实时聊天的经验性理解。基于来自 Gitter 上八个热门社区的 173,278 个对话,我们探索了开发人员的时间通信概况、社交网络及其对社区的属性、讨论主题的分类以及实时聊天中的交互模式。我们的研究揭示了一些有趣的发现。此外,我们为 OSS 开发人员和社区提供建议,突出在线通信平台供应商的高级功能,并为 OSS 研究人员提出富有洞察力的未来研究问题。未来,我们计划研究如何将对话自动分类为不同的主题,并尝试根据已回答的问题及其来自实时聊天的相应解决方案构建知识库。我们希望我们发现的发现和见解能为其他研究铺平道路,有助于推动对 OSS 开发协作的更深入理解,并促进对海量聊天记录中嵌入的知识的更好利用和挖掘。为了便于复制或其他类型的未来工作,我们在线提供了本研究中使用的对话数据和解开的对话: <https://github.com/LiveChat2021/LiveChat>。

致谢

我们非常感谢匿名审稿人对改进这份手稿提出的建设性和有见地的建议。这项工作得到了中国国家重点研发计划项目的支持,项目编号为 2018YFB1403400,国家自然科学基金项目项目编号为 61802374、62002348 和 62072442,以及中国科学院青年创新促进会。

ESEC/FSE 21,2021 年 8 月 23 日至 28 日,希腊雅典

参考

[1]原型和实验 Angular 代码的工具。2020 年。 <http://plnkr.co/>。

[2]爆炸人工智能。2020. 太空。 <https://spacy.io/>。

[3] Miltiadis Allamanis 和查尔斯·萨顿。2013. 原因,时间和内容:按主题,类型和代码分析堆栈溢出问题,在第十届采矿软件存储库工作会议论文集,MSR 13. IEEE 计算机学会,53-56. <https://doi.org/10.1109/MSR.2013.6624004> [4] Deeksha Arya, Wenting Wang, Jin LC Guo, and Jinghui Cheng. 2019.开源软件问题讨论的信息类型分析与检测。

在第 41 届国际软件工程工作会议论文集,ICSE 2019,454-464. <https://doi.org/10.1109/ICSE.2019.00058>

[5] Mehdi Bagherzadeh 和 Raffi Khatchadourian。2019. 走向大:关于大数据开发人员要求的大规模研究。在欧洲软件工程会议ACM 联合会议和软件工程基础研讨会论文集上,ESEC/SIGSOFT FSE 2019.ACM,432-442. <https://doi.org/10.1145/3338906.3338939>

[6]马修·巴斯蒂安·塞巴斯蒂安·海曼和马修·雅科米。2009. Gephi:用于探索和操作网络的开源软件。在国际 AAAI 网络和社交媒体会议论文集上,卷。3。

[7]亚历克斯·巴韦拉斯。1950. 面向任务的小组中的沟通模式。美国声学学会杂志 22, 6 (1950), 725-730。

[8] S. Beyer,C. Macho,M. Di Penta 和 M. Pinzger。2017. 分析Android API 类及其在 Stack Overflow 上的引用之间的关系。技术报告。克拉根福大学,慕尼黑大学。

[9] Stefanie Beyer,Christian Macho,Martin Pinzger 和 Massimiliano Di Penta。2018 年。在 Stack Overflow 上自动将帖子分类为问题类别。在第 26 届程序理解会议记录中,JCPC 2018.ACM, 211-221. <https://doi.org/10.1145/3196321.3196333> [10]斯蒂芬妮·拜尔和马丁·平茨格。2014. A Manual Categorization of Android App Development Issues on Stack Overflow.在第 30 届 IEEE软件维护与演进国际会议上。IEEE 计算机协会,531-535. [HTTPS://doi.org/10.1109/ICSME.2014.88](https://doi.org/10.1109/ICSME.2014.88) [11] Christian Bird,Alex Gourley,Prem Devanbu,Michael Gertz 和 Anand Swami nathan。2006. 挖掘电子邮件社交网络。在 2006 年国际挖掘软件存储库研讨会论文集上。137-143. <https://doi.org/10.1145/1137983.1138016>

[12] A. Bosu,CS Corley,D. Heaton,D. Chatterji,JC Carver 和 NA Kraft。2013. 在 StackOverflow 中建立声誉:实证调查。2013年第十届挖掘软件存储库 (MSR)工作会议。89-92. <https://doi.org/10.1109/MSR.2013.6624013>

[13] Jérémie Bouttier,Philippe Di Francesco 和 Emmanuel Guitter。2003. 平面图中的测地线距离。核物理学 B 663, 3 (2003), 535-567。

[14]法比奥·卡莱法托·菲利波·拉努比莱和妮可·诺维埃利。2018. 如何寻求技术帮助?在 Stack Overflow 上写问题的循证指南。信息。软件。技术。94 (2018), 186-207. <https://doi.org/10.1016/j.infsof.2017.10.009>

[15] Preetha Chatterjee,Kostadin Damevski,Nicholas A. Kraft 和 Lori L. Pollock。2020. 与软件相关的 Slack Chats with Disentangled Conversations.在 MSR 20:第 17 届国际挖掘软件存储库会议中。ACM,588-592. <https://doi.org/10.1145/3379597.3387493> [16] 微软公司。2020. 节点。 <https://nodejs.org/en/>。

[17] 微软公司。2020.打字稿。 <https://www.typescriptlang.org/>。

[18] Fabio Crestani 和 Mounia Lalmas。2001. 信息检索中的逻辑和不确定性。在信息检索讲座,计算机科学讲义。施普林格出版社。 https://doi.org/10.1007/3-540-45368-7_9 [19] Jacob Devlin,Ming-Wei Chang,Kenton Lee 和 Kristina Toutanova。2019. BERT:用于语言理解的深度双向变压器的预训练。在计算语言协会北美分会 2019 年会议论文集:人类语言技术,NAACL-HLT 2019,第 1 卷 (长篇和短篇论文)。计算语言协会, 4171-4186. <https://doi.org/10.18653/v1/n19-1423>

[20] 莱因哈德·迪斯特尔。2005. 图论。2005 年毕业。数学 101 (2005) 中的文本。

[21] 公司码头工人。2020. 码头工人。 <https://www.docker.com/>。

[22] 莎莉芬奇和乔什 D. Tenenberg。2005. 理解卡片分类数据。专家系统。J.诺尔。英。22, 3 (2005), 89-93. <https://doi.org/10.1111/j.1468-0394.2005.00299.x>

[23]布赖恩·菲茨杰拉德。2006. 开源软件的转型。问题 30, 3 (2006), 587-598。

[24] 以太坊基金会。2020. 以太坊。 <https://ethereum.org/en/>。

[25] JS 基金会。2020 年。 <http://appium.io/>。

[26]林顿·C·弗里曼。1977. 一套基于介数的中心性度量。社会计量学 (1977) 35-41。

[27] 亚当·吉布森。2020. 深度学习 4j。 <https://deeplearning4j.org/>。

[28] 吉特。2020. REST API。 <https://developer.gitter.im/docs/rest-api>。

[29] Jesús M. González-Barahona 和 Gregorio Robles。2013. 自由,自由,开源软件社区的趋势:从志愿者到公司/Ak tuelle 自由,自由和开源软件Gemeinschaften 的趋势:Von Freiwilligen zu Unternehmen。它Inf.技术。55, 5 (2013), 173-180. [HTTPS://doi.org/10.1016/j.infsof.2013.05.001](https://doi.org/10.1016/j.infsof.2013.05.001)

林石, 萧晨,叶阳,姜寒之,姜子友,南牛,王青

<https://doi.org/10.1524/itit.2013.1012>

[30] 谷歌。2020. 角。 <https://angular.io/>。

[31] 谷歌。2020.吉特。 <https://gitter.im/>。

[32] 谷歌。2020. 松弛。 <https://slack.com/>。

[33]郭高阳,王超坤,陈俊,葛鹏程。2018. 谁在回答谁?在具有长短期记忆网络的群聊中找到“回复”关系。在第 七届新兴数据库国际会议论文集上。 <https://doi.org/10.1007/s10586-018-2031-4> [34] Anja Guzzi, Alberto Bacchelli,Michele Lanza,Martin Pinzger 和 Arie van Deursen。2013. 开源软件开发邮件列表中的通信。在第 10 届挖掘软件存储库工作会议论文集上,MSR 13. IEEE 计算机学会,277-286. <https://doi.org/10.1109/MSR.2013.6624039>

[35]韩俊晓,Emad Shihab,万志远,邓水光,夏心。2020 年。程序员如何讨论深度学习框架。帝国。软件。英。25, 4 (2020), 2694-2747. <https://doi.org/10.1007/s10664-020-09819-6> [36]黄乔、夏心,罗大卫和盖尔 C. 墨菲。2018. 自动化意图挖掘。IEEE 软件工程汇刊 PP,99 (2018 年) 1-1. [HTTPS://doi.org/10.1109/ICSM.2015.7332474](https://doi.org/10.1109/ICSM.2015.7332474) [37] Verena Käfer,Ivan Bogicevic,Stefan Wagner 和 Jasmin Ramadanli。2018.开源项目中的通信 电子邮件时代的终结?在第 40 届国际软件工程会议论文集:配套论文集, ICSE 2018.ACM,242-243. <https://doi.org/10.1145/3183440.3194951> [38]瓦尔德斯·克雷布斯和琼·霍利。2004. 通过社交网络发展建设可持续社区。非营利季刊 11 (01 2004), 46-53。

[39] Jonathan K. Kummerfeld,Sai R. Gouravajhala,Joseph Peper,Vignesh Athreya, Chulaka Gunasekara,Jatin Ganhotra,Siva Sankalp Patel,Lazaros Polymenakos 和 Walter S. Lasecki。2019. 一个大规模的对话解释语料库。在计算语言协会第 57 届年会论文集 (第 1 卷:长论文)。 <https://doi.org/10.18653/v1/p19-1374> [40] Bin Lin, Alexey Zagalsky,Margaret-Anne D. Storey 和 Alexander Serebrenik。2016. 开发人员为何懈怠:了解软件团队如何使用 Slack。在第 19 届 ACM 计算机支持的合作工作和社会计算会议论文集。333-336. <https://doi.org/10.1145/2818052>。

2869117

[41]刘慧,施展,顾家臣,刘全,思伟,朱小丹。2020.基于端到端过的在线对话解开。在第二十九届国际人工智能联合会议论文集上,JJCAI 2020。jjcai.org,3868-3874. <https://doi.org/10.24963/jjcai.2020/535>

[42] 剧团科技有限公司。2020. 吉特。 <https://gitter.im/>。

[43]姚璐,毛新军,周明辉,张扬,王涛,李祖德。2020 年。勿忙造成浪费:堆栈溢出中快速答案的实证研究。在IEEE 国际软件维护 and 演进会议上,ICSME 2020。IEEE,23-34. <https://doi.org/10.1109/ICSME46990.2020.00013> [44] Manju, K., Ahuja, Kathleen, M., and Carley。1998. 虚拟组织中的网络结构。计算机中介通信杂志 (1998 年)。 [HTTPS://doi.org/10.1111/j.1083-6101.1998.tb00079.x](https://doi.org/10.1111/j.1083-6101.1998.tb00079.x)

[45] Andrew Meneely,Laurie A. Williams,Will Snipes 和 Jason A. Osborne。2008 年。使用开发人员网络和社交网络分析预测故障。在第 16 届 ACM SIGSOFT软件工程基础国际研讨会论文集上,2008 年.ACM,13-23. <https://doi.org/10.1145/1453101.1453106> [46]考特尼·米勒·佩奇·罗德盖罗·玛格丽特·安妮·D·斯托里·德娜·福特和托马斯·齐默尔曼。2021. “你周末过得怎么样?” COVID-19 期间在家工作的软件开发团队。CoRR abs/2101.05877 (2021)。

[47] Ana Moreira,Awais Rashid 和 João Araújo。2005. 需求工程中关注点的多维分离。在第 13 届 IEEE 国际需求工程会议 (RE 2005)上。IEEE 计算机学会,285-296. [HTTPS://doi.org/10.1109/RE.2005.46](https://doi.org/10.1109/RE.2005.46) [48] Sebastiano Panichella,Andrea Di Sorbo,Emिता Guzman,Corrado Aaron Visag gio,Gerardo Canfora 和 Harald C Gall。2015. 如何改进我的应用程序?对软件维护和演进的用户评论进行分类。(2015), 281-290。

[49]埃斯特班·帕拉·阿什利·埃利斯和索尼娅·海杜克。2020. GitterCom: Gitter 中的开源开发人员通信数据集。在 MSR 20:第 17 届国际挖掘软件存储库会议中。ACM,563-567. <https://doi.org/10.1145/3379597.3387494> [50]陈曲,刘洋,W. Bruce Croft,张永峰,Johanne R. Trippas 和 Minghui Qiu。2019. 信息搜索对话中的用户意图预测。

在 2019 年人类信息交互和检索会议记录中,CHIIR 2019.ACM,25-33. <https://doi.org/10.1145/3295750.3298924> [51]克里斯托弗·罗森和埃马德·希哈布。2016. 移动开发者在问什么?使用堆栈溢出的大规模研究。帝国。软件。英。21, 3 (2016), 1192-1223. <https://doi.org/10.1007/s10664-015-9379-3>

[52]戈登·鲁格和彼得·麦乔治。2005. 分类技术:关于卡片分类、图片分类和项目分类的教程论文。专家系统。J.诺尔。英。22, 3 (2005), 94-107. <https://doi.org/10.1111/j.1468-0394.2005.00300.x> [53] Khadijah Al Safwan 和弗朗西斯科·什人。2019.分解代码提交的基本原理:软件开发人员的观点。在 ACM 会议记录中

开发人员在 Gitter 上的实时聊天初探

ESEC/FSE 21,2021年8月23日至28日,希腊雅典

欧洲软件工程会议和软件工程基础研讨会联席会议,ESEC/SIGSOFT FSE 2019,ACM, 397-408. <https://doi.org/10.1145/3338906.3338979>

[54] Jorge M. Santos 和 Mark J. Embrechts. 2009.关于使用调整后的兰德指数作为评估监督分类的指标.在人工神经网络 - ICANN 2009 年第 19 届国际会议上,利马索尔 (计算机科学讲义,第 5769 卷) 。施普林格,175-184. https://doi.org/10.1007/978-3-642-04277-5_18

[55] Roland Robert Schreiber 和 Matthäus Paul Zylka. 2020.软件开发项目中的社交网络分析:系统文献综述.诠释. J.软件。

英.知道.英. 30, 3 (2020), 321-362. <https://doi.org/10.1142/s021819402050014x>

[56] 窦申,杨强,孙建涛,陈郑. 2006. 动态文本消息流中的线程检测.在 SIGIR 2006:第 29 届国际 ACM SIGIR 信息检索研究与开发会议论文集. ACM,35-42. <https://doi.org/10.1145/1148170.1148180>

[57]林石,邢明哲,李明阳,王亚文,李守斌,王庆。

2020. 通过深度连体网络检测来自海量聊天消息的隐藏特征请求.在 ICSE 20:第 42 届软件工程国际会议上. ACM,641-653. <https://doi.org/10.1145/3377811.3380356>

[58] Emad Shihab,Zhen Ming Jiang 和 Ahmed E. Hassan. 2009. 关于GNOME GTK+ 项目的开发人员使用 Internet 中继聊天 (IRC) 会议。

在第 6 届国际挖掘软件存储库工作会议论文集上,MSR 2009 (与 ICSE 合用) 。 107-110. <https://doi.org/10.1109/MSR.2009.5069488>

[59] Emad Shihab,Zhen Ming Jiang 和 Ahmed E. Hassan. 2009. 研究在开源项目中使用开发人员 IRC 会议.在第 25 届 IEEE 国际软件维护会议 (ICSM 2009) 上. 147-156. <https://doi.org/10.1109/ICSM.2009.5306333>

[60] Andrea Di Sorbo,Sebastiano Panichella,Corrado Aaron Visaggio,Massimiliano Di Penta,Gerardo Canfora 和 Harald C. Gall. 2015. 开发电子邮件内容分析器:开发人员讨论中的意图挖掘 (T).在

第 30 届 IEEE/ACM 自动化软件工程国际会议,ASE 2015,12-23. <https://doi.org/10.1109/ASE.2015.12>

[61]亚历山大·斯特雷尔和乔伊迪普·戈什. 2002. Cluster Ensembles 组合多个分区的知识重用框架. J.马赫.学.水库. 3 (2002), 583-617。

[62] Neny Sulistianingsih 和 Edi Winarko. [未].社交网络数据集中分子复合物检测的性能分析.国际计算机应用杂志 975 ([nd]) ,8887。

[63]谭鑫,周明辉,孙泽宇. 2020 年.在 GitHub 上先看看好的第一个问题.在 ESEC/FSE 20:第 28 届 ACM 联合欧洲软件工程会议和软件工程基础研讨会上,虚拟活动,美国,2020 年 11 月 8 日至 13 日,Prem Devanbu,Myra B. Cohen 和 Thomas Zimmermann (主编) . ACM, 398-409. <https://doi.org/10.1145/3368089.3409746>

[64] Christoph Treude,Ohad Barzilay 和 Margaret-Anne D. Storey. 2011. 程序员如何在 Web 上提问和回答问题?在第 33 届国际软件工程会议论文集上,ICSE 2011,ACM,804-807. <https://doi.org/10.1145/1985793.1985907>

[65]斯坦利·沃瑟曼和凯瑟琳·浮士德. 1994. 社交网络分析:方法和应用.剑桥大学出版社. <https://doi.org/10.1017/CBO9780511815478>

[66]邓肯 J 瓦茨和史蒂文 H Strogatz. 1998. “小世界”网络的集体动力.自然 393, 6684 (1998), 440-442。

[67]于立国,斯里尼·拉马斯瓦米,阿洛克·米什拉和迪普蒂·米什拉. 2011. 全球软件开发中的通信:使用 GTK+ OSS 存储库的实证研究.在 2011 年关于转向有意义的利益系统的国际联合会议的论文集中,OTM 11. 218-227. https://doi.org/10.1007/978-3-642-25126-9_32

[68]张玉霞,周明辉,克拉斯-扬·斯托尔,吴建宇,金志. 2020.公司如何在开源生态系统中合作? : OpenStack 的实证研究.在 ICSE 20:第 42 届软件工程国际会议上。

ACM,1196-1208. <https://doi.org/10.1145/3377811.3380376>