

基于知识图谱的 bug 问题探索性搜索方法

孙小兵^{1,2}, 王璐¹, 王经纬¹, 李斌¹, 李宇¹

(1. 扬州大学信息工程学院, 江苏扬州 225127; 2. 南京大学计算机软件新技术国家重点实验室, 江苏南京 210023)

摘要: 软件 bug 问题在软件开发与维护过程中不可避免. 然而软件历史库中的 bug 库与 commit 库之间没有直接的对应关系, 并且随着 bug 信息、commit 提交信息量的增长, 搜索 bug 变得更加繁琐. 本文提出一种基于知识图谱对 bug 问题进行探索性搜索的方法. 通过建立 bug 报告、commit 提交信息及相关人员 (如: bug 报告人、commit 提交者等) 信息的 bug 知识图谱, 并结合探索性搜索的思想, 不仅能准确搜索 bug 问题, 还能提供相关辅助信息对 bug 问题进行探索性搜索, 帮助软件开发人员更有效地理解和解决 bug 问题.

关键词: bug 报告; commit 提交; 知识图谱; 探索性搜索

中图分类号: TP311

文献标识码: A

文章编号: 0372-2112 (2018)07-1578-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2018.07.006

Construct Knowledge Graph for Exploratory Bug Issue Searching

SUN Xiao-bing^{1,2}, WANG Lu¹, WANG Jing-wei¹, LI Bin¹, LI Yu¹

(1. School of Information Engineering, Yangzhou University, Yangzhou, Jiangsu 225127, China;

2. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, China)

Abstract: Software bug issues are inevitable in software development and maintenance. However, there are no corresponding relationships between bugs and commits in software repository. Moreover, with the increasing amount of bug reports and commit information, bug search in the software repository becomes more difficult and costly. In this paper, we propose an exploratory search approach to search bug issues based on knowledge graph. By building the bug knowledge graph of bug reports, commits and related developers (such as bug reporters, committers and so on) and combining with the idea of exploratory search, our approach can not only help software developers search bug issues accurately, but also provide the relevant information to explore bug issues, thus software developers can understand and resolve bug issues more effectively.

Key words: bug report; commit; knowledge graph; exploratory search

1 引言

在软件开发和维护过程中, 软件开发人员和维护人员常常会遇到一些 bug 问题^[1]. 要解决这些 bug 问题, 开发人员常常参照与该 bug 报告对应的 commit 记录^[2,3]或通过搜索历史库中相似的已被报告并且被解决的 bug 来寻找解决方案^[4,5]. 而已有的 bug 库或 commit 库大多基于关系数据库进行搜索, 得到的信息较为片面, 只能搜索 bug 信息, 并不能直接提供相关的 commit 信息或者相似的 bug 信息. 且这种搜索方式不能理解开发人员的搜索目的, 传统的搜索方式都是基于关键字匹配进行搜索的, 当开发人员输入的搜索内容不

够明确时, 得到的搜索结果并不准确. 并且软件历史库中的 bug 报告与 commit 信息间并没有建立相互对应的关联关系, 这也使得当开发人员对 bug 问题进行搜索时, 还要花时间去查找与之对应的 commit 信息. 随着 bug 报告和 commit 信息量的不断增长, 搜索也会越来越繁琐耗时.

鉴于 bug 报告与 commit 之间的这种关联关系的重要性, 很多研究提出了一些方法来建立 bug 报告和 commit 之间的这种关联关系, 或者尝试找出与之相似的 bug. Wu 等人提出一种方法 ReLink^[6], 根据计算 bug 报告和 commit 信息间的文本相似度来判断它们之间的关系, 文本间的相似度越高, 此 commit 就越有可能与该

收稿日期: 2016-12-26; 修回日期: 2017-08-21; 责任编辑: 孙瑶

基金项目: 国家自然科学基金 (No. 61402396, No. 61472344); 南京大学计算机软件新技术国家重点实验室开放基金 (No. KFKT2016B21); 江苏青蓝工程; 大学生创新创业训练计划 (No. 201611117024Z); 扬州市自然科学基金-面上项目 (No. YZ2017113)

bug 报告之间存在关联关系. 此外相似 bug 报告很大程度上也能帮助软件的开发者解决 bug 问题, 软件开发者可以通过类似的 bug 对自己的问题有更深入的了解, 也可以通过类似 bug 的 commit 寻找解决方案. Gu 等人为了搜索类似的 bug, 以执行轨迹的形式开发了语义 bug 信息, 它能够精确捕获 bug 语义^[7]. 然而, 这些方法虽好, 想要获得全面的 bug 相关信息, 还需将这两种方法结合起来, 获取到 bug 相对应的 commit 信息及相似的 bug 信息才能更好的帮助软件开发人员解决 bug 问题.

为了更好的帮助软件开发人员理解和解决 bug 问题, 本文提出一种基于知识图谱对 bug 问题进行探索性搜索的方法^[8]. 将知识图谱应用于 bug 搜索不仅可以为开发者提供更全面的 bug 相关信息, 还能改进搜索质量. 本方法首先将 bug 报告、commit 及开发人员的数据使用自然语言处理技术^[9]进行处理, 并以知识图谱的数据模式建立 bug 知识库, 并得到 bug 及其相关信息的知识图谱. 然后使用自然语言预处理技术对开发者的查询输入进行语义处理, 结合探索性搜索方法, 不仅能准确提供要搜索的内容, 还能在让开发者在与界面不断交互的过程中帮助他们与 bug 及其相关信息进行探索和学习, 从而辅助开发者更快地解决 bug 问题.

2 研究背景和相关工作

2.1 基于知识图谱的搜索方式

知识图谱 (Knowledge Graph) 由 Google 正式提出, 其目的是为了改善搜索结果质量. 传统的信息检索一般基于关键词进行检索, 但是搜索引擎并不理解用户的输入, 只能通过对开发者的输入进行切分得到关键词, 然后将关键词与要搜索的数据进行匹配, 再将匹配的结果通过一定的排序算法返回, 最终搜索者从中浏览选择想要的搜索结果. 因为不能理解搜索者的真实目的, 所以基于关键词的检索技术效果不是很好. 而基于知识图谱的语义检索可理解用户的输入, 并为搜索者提供直接和系统的结果^[10-12].

一个典型的知识图谱包括实体、概念和关系^[13]. 知识图谱的目标在于描述现实世界中存在的各种实体或概念, 以及这些实体或概念之间的关联关系. 在知识图谱中, 不仅概念与概念之间存在关联关系, 实体与实体之间也存在关联关系, 实体可以拥有属性, 用于刻画实体的内在特性. 知识图谱是一种可以用来有效管理大规模数据的工具, 可应用在语义搜索、知识导航、数据分析及决策支持等领域^[14].

当前关于知识图谱的研究包括对结构化数据和半结构化数据的挖掘与分析, 而大多研究集中在结构化的数据, 例如基于成熟的知识库^[10] (如 Freebase^[15]). 而本文 Mozilla@ Bugzilla^① 和 Github^② 中的 bug 数据和 com-

mit 数据是半结构化的, 并且也没有成熟的 bug 知识库可利用. 因此, 通过对数据的分析, 将 bug 报告、commit 信息及开发人员信息建立知识图谱数据模式的数据库, 并生成 bug 知识图谱, 不仅建立了数据间的结构化关系, 也可更好地理解开发人员的查询输入, 有助于为他们提供更加全面的搜索结果.

2.2 探索性搜索

探索性搜索是一种通过搜索进行知识的获取、概念的理解、数据的比较以及对信息的分析和合成等为目的的搜索活动^[16]. 为了让用户参与到搜索过程中来, 可通过高度交互的界面, 使用户处于持续的控制之中^[16], 通过动态选择来探索并预测用户的搜索意图, 并记录搜索记录, 浏览时间等, 对这些数据进行挖掘, 以提供更好的服务, 然后根据搜索者对搜索结果的采纳情况改变结果的排序, 优化搜索结果. 在本文方法中, 探索性搜索是基于建立的知识图谱进行的, 搜索过程中可通过对信息的动态选择, 不断地帮助优化搜索结果, 同时也提供更全面的信息帮助搜索者进行探索和学习, 发现信息间的关联关系.

2.3 相关工作

一些研究提出了用于恢复 bug 报告和它们相应的 commit 之间链接的技术. Wu 等人提出一种方法 ReLink^[6], ReLink 基于三个标准恢复链接关系, 包括 commit 信息和 bug 报告间文本相似度、bug 报告的提交和代码修改的 commit 提交之间的持续时间、以及 bug 报告的评论者和提交者之间的映射. Le 等人提出一种新的 bug 链接方法 RCLinker^[2]来解决当 commit 描述信息为空或只包含较少的信息时难以恢复 bug 报告与 commit 提交间链接的问题. 他们依赖丰富的上下文信息来检测 bug 报告和 commits 间的关系. 并且他们还提出一种新的基于分类技术来建立判别模型预测 bug 报告和 commit 间的链接是否存在.

另外, 当前也有一些研究工作致力于通过捕获 bug 的语义, 寻找相似的 bug 来解决遇到的 bug 问题. Gu 等人描述了一种 Eclipse 的插件 OSCILLOSCOPE, 它使用 bug 追踪来详尽地搜索它的数据集, 从而找出相似的 bug 并返回它们的 bug 报告^[4].

本文提出的搜索方法是通过建立 bug 报告与 commit 及相关开发人员的 bug 知识图谱, 不仅恢复 bug 与 commit 间关系, 还能提供更全面的相关信息, 更好的帮助搜索 bug 以达到对 bug 相关信息进行探索性理解和搜索的目的.

3 基于知识图谱的探索性搜索

图 1 给出了本文技术的流程图, 分为两个阶段, 第

① <https://bugzilla.mozilla.org/>

② <https://github.com/github>

1 阶段使用自然语言处理技术对输入的搜索内容进行自然语言处理,从处理后的内容中识别出实体,概念和属性(本文将 bug 的描述信息作为该 bug 的实体).然后结合 bug 报告、commit 及开发人员的知识图谱的数据模式,对自然语言处理的结果与知识库^[17]中的描述信息进行聚类^[18],后进行相似度的计算,并依据相似度排序将相应的 bug 信息返回开发者.而第 2 阶段则根据第 1 阶段得到的 bug 搜索结果,通过在交互界面中对其相关信息进行动态选择,了解更全面的信息,从而完成探索性搜索,找出 bug 的解决方法.

3.1 基于知识图谱对 bug 问题的搜索

bug 报告、commit 提交数据大多是半结构化的数据,而要建立知识图谱并没有一些现有的知识库可利用,且 bug 数据相互间的层次关系也不明显,因此要构建 bug 报告、commit 及开发人员的知识图谱,先要构建基于知识图谱数据模式的 bug 知识库^[19],后建立对应知识图谱并将其可视化显示.本文知识图谱中的实体定义为:bug 的描述信息、commit 的描述信息、人员的姓名;将其中的关系定义为:bug 与 bug 间的 duplicates、depends on 关系及 bug 间的相似关系、commit 与 bug 间的对应关系、commit 与 commit 之间的相似关系以及人员与 bug 间的对应关系、人员与 commit 之间的对应关系.

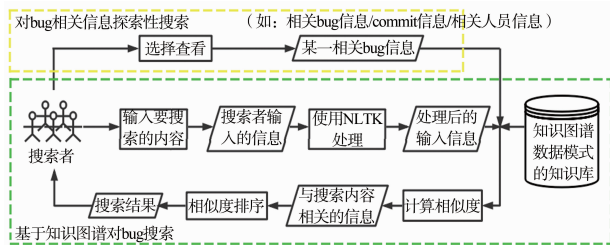


图1 基于知识图谱对bug问题探索性搜索流程图

3.2 对 bug 相关信息的探索性搜索

本节主要介绍基于知识图谱对 bug 问题探索性搜索的第 2 阶段的内容,包括在探索搜索过程中涉及到的标签云的生成方法,以及如何借助知识图谱和标签云帮助开发者对 bug 及其相关信息进行探索性搜索^[20-23].

3.2.1 标签云的生成

当提取出相关的搜索结果后,本方法将与输入的信息最相似的搜索结果使用 TextRank^[14]自动摘要算法对其在聚类结果中所处的类别中的描述信息进行关键词的提取操作,将该类别中的描述信息拆分成关键词,生成标签云,返回给开发者.

3.2.2 通过交互界面进行探索性搜索

为了使开发者更好地参与到搜索过程中,搜索界面设置了一些选项,当开发者对搜索的信息有一定的了解时,可先对要搜索的信息类别进行选择,若要搜索

的是 bug 的信息,可先进行:产品(product)、平台(platform)、组件(component)这三个方面的筛选,缩小要处理的信息范围,加快数据的处理,后输入对信息的描述,使得到的搜索结果更加准确.当开发者对要搜索的信息不是很明确时,输入的描述信息不够详细规范,因此返回的搜索结果信息量较多,按照相似度从高到低的排序呈现搜索结果.

当开发者查看搜索结果时,每点击查看一个搜索结果,就会显示与这个搜索结果相关的信息.如:当开发者查看某 bug 信息时,就会显示与之相关的 bug 信息、类似的 bug 信息、以及与之相关联的 commit 信息和开发人员信息,甚至还能将当前搜索得到的相似度高的 bug 信息中的描述进行关键词提取,并以标签云的形式显示,供开发者查看是否与自己要搜索的 bug 内容相关,也可点击进行动态选择,进一步进行信息的探索,这不仅能使开发者对要搜索的信息有更全面的认识,还能帮助开发者对搜索到的结果进行比较和判断,直到最终准确找到要搜索的信息.

4 实验评估

4.1 实验设置

本文方法已使用 Java 编程语言进行实现.实验邀请 20 人参与实验评估.每位参评者分别使用本文实现的工具和 Mozilla@ Bugzilla 平台搜索 bug 问题,通过统计返回的前 50 条数据中的相关数据量计算搜索结果的查准率,以此验证本文提出的搜索方法的有效性.

实验研究主要包括两个方面.首先,根据查准率来衡量搜索结果中相关信息的相关性.其次,研究本文提出的方法提供的搜索结果是否确实能够有效地帮助开发者解决 bug 问题.基于上述两个方面的研究目标,设计以下 3 个研究问题.

问题 1 通过本文的搜索方法搜索得到的相关信息是否有帮助?

问题 2 与现有的基于 bug 平台和 commit 平台搜索 bug 的方法相比,本文的方法是否有效?

问题 3 本文的方法提供的信息是否能有效帮助开发人员解决 bug 问题?

4.2 数据集

本实验从 Mozilla@ Bugzilla 及 Github 中爬取了 bug 报告、对应的 commits 及开发人员信息,共有 22 万条 bug 数据,2.5 万条 commit 数据,以及 1.2 万条人员数据. Mozilla@ Bugzilla 是一个开源的针对 Mozilla 项目中的产品的 bug 跟踪系统,具有强大的检索功能和强大的后端数据库支持,它用于跟踪和管理 bug. Github 是一个代码托管平台和开发者社区,开发者可以在 Github 上创建自己的开源项目并与其他开发者协作编码.我

们将这些数据处理生成了 bug、commit 及人员信息的知识图谱并将其应用在本文的搜索方法中。

4.3 评测指标

本文通过计算查准率 (Precision) 来衡量本文方法的有效性。该度量基于两项数据统计结果来计算:分别是 True Positives (TP) 和 False Positives (FP), 定义如下:

True Positives: 检索到的相关的信息数量。

False Positives: 检索到的不相关的信息数量。

通过使用下面的公式来计算查准率:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4.4 结果与分析

问题 1 当参评者使用本文工具对 bug 问题进行搜索后,通过查看这些搜索信息,按四个等级(很有用、有用、一般有用、无用)判断这些信息是否帮助解决 bug 问题。结果显示在使用本文工具进行搜索返回的相关信息中,有 35% 的人认为提供的相关信息是很有用的,能够帮助解决 bug 问题;45% 的人认为提供的相关信息是有用的,能提供相关信息对其进行了解和跟进;还有 20% 的人认为提供的相关信息一般有用,并不能有效帮助解决 bug 问题,但可以帮助了解 bug。因此本文提出的搜索方法提供的相关信息大部分是有用的,可帮助开发人员了解要解决的 bug 甚至能帮助解决 bug 问题。

问题 2 参评者分别在本文实现工具和 Mozilla@ Bugzilla 中选择相同的产品类型,并输入相同的搜索内容,进行 bug 搜索,我们通过统计参评者对前 50 条返回的结果中的 bug 信息的相关性判断,计算其查准率。表 1 显示了每位参评者分别使用我们的工具及 Mozilla@ Bugzilla 搜索相同问题的查准率详细比较情况。由表 1 可知,使用 Mozilla@ Bugzilla 的平均查准率为 13.52%,而使用本文工具的查准率为 56.78%。由此可知,本文工具的搜索结果明显优于使用 Mozilla@ Bugzilla 搜索的结果。

问题 3 本文通过统计调查参评者分别使用本文工具以及 Mozilla@ Bugzilla 来解决未被解决的 bug 问题的比率来验证本文方法实现的工具是否能有效解决 bug 问题。使用本文工具对 bug 问题进行搜索时,其中有 35% 的 bug 能够得到解决。使用 Mozilla@ Bugzilla 对 bug 问题进行搜索时,其中有 15% 的 bug 能够得到解决。这一结果表明,本文工具相比 Mozilla@ Bugzilla 具有一定的优势,且确实能够帮助开发者解决 bug 问题。但由于本文实验中爬取到的实验数据量的局限性,本文只给一小部分的 bug 数据匹配了相关的 commit,还有很多的 bug 没有相对应的 commit 信息提供给参与人员。在今后研究中会进一步对其完善,提高相关信息数量,从而更好的帮助开发者解决 bug 问题。

表 1 我们的工具与 Mozilla@ Bugzilla 查准率比较

序号	参评者身份	要搜索的 bug	输入搜索内容	我们工具的查准率	Mozilla@ Bugzilla 查准率
1	老师 1	bug 590468	Reduce size of chrome document layer due to status bar	78.26%	50%
2	老师 2	bug 105842	Empty or no < title > or image does not update title in tabbed browsing	70.83%	30%
3	开发者 1	bug 538103	font fullnames (e.g. Arial Bold) are matched incorrectly as font family names	65.21%	25%
4	开发者 2	bug 137155	POST request sent as two small packets (IIS 5 sometimes chokes)	86.48%	16.67%
5	开发者 3	bug 357197	OCSP response code fails to match CERTIds	79.16%	28.57%
6	开发者 4	bug 306087	no Blizzard GTKWidget after building Minimo	78.57%	0
7	学生 1	bug 414287	Share search result processing logic as AutoCompleteProcessSearch	62.50%	27.78%
8	学生 2	bug 373175	Avoid arena naming code when JS_ ARENAMETER is not defined	41.67%	0
9	学生 3	bug 346356	Replace deprecated calls to preventBubble() on trunk	25.00%	33.33%
10	学生 4	bug 338167	Localizability issues with recurNthWeek and recurLast in dateFormat. properties	33.33%	0
...
平均查准率				56.78%	13.52%

5 结论与展望

本文基于知识图谱和探索性搜索的思想提出了一种用于搜索和解决 bug 问题的新方法. 实验结果说明了本文的搜索方法与现有的搜索方式相比, 不仅搜索结果更为准确, 能够提供更全面的信息, 且能帮助开发者更好的解决 bug 问题.

但是本文方法中依然还存在一些问题有待进一步解决. 对于数据库更新方面, 存在着更新时间会随着数据量的增加而越来越长的问题, 拟使用在线的终身学习模型来实现 bug 数据的自动更新. 另外, 进一步完善 bug 知识图谱从而更好的帮助开发者解决问题^[24,25].

参考文献

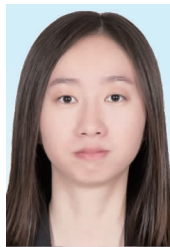
- [1] 李勇, 黄志球, 王勇, 等. 数据驱动的软件缺陷预测研究综述[J]. 电子学报, 2017, 45(4): 982-988.
LI Yong, HUANG Zhi-qiu, WANG Yong, et al. Survey on data driven software defects prediction[J]. Acta Electronica Sinica, 2017, 45(4): 982-988. (in Chinese)
- [2] Le T D B, Linares-Vásquez M, Lo D, et al. RCLinker: automated linking of issue reports and commits leveraging rich contextual information[A]. Proceedings of IEEE 23rd International Conference on Program Comprehension (ICPC) [C]. USA: IEEE, 2015. 36-47.
- [3] 孙小兵, 刘湘月, 李斌, 等. 基于相关主题模型的程序网络自动构建与分析[J]. 电子学报, 2017, 45(5): 1052-1056.
SUN Xiao-bing, LIU Xiang-yue, LI Bin, et al. On automatic construction and analysis of program network via relational topic model [J]. Acta Electronica Sinica, 2017, 45(5): 1052-1056. (in Chinese)
- [4] Gu Z, Barr E T, Schleck D, et al. Reusing debugging knowledge via trace-based bug search [J]. ACM SIGPLAN Notices, 2012, 47(10): 927-942.
- [5] Sun X, Li Y, Li B, Chen Y. What information in software historical repositories do we need to support software maintenance tasks? an approach based on topic model [A]. Computer and Information Science, Studies in Computational Intelligence (Volume 566) [M]. Berlin: Springer, 2015. 27-37.
- [6] Wu R, Zhang H, Kim S, et al. Relink: recovering links between bugs and changes[A]. Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering [C]. USA: ACM, 2011. 15-25.
- [7] Bachmann A, Bird C, Rahman F, et al. The missing links: bugs and bug-fix commits [A]. Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering [C]. USA: ACM, 2010. 97-106.
- [8] Wang L, Sun X, Wang J, et al. Construct bug knowledge graph for bug resolution: poster [A]. Proceedings of the 39th International Conference on Software Engineering Companion [C]. USA: IEEE, 2017. 189-191.
- [9] Sun X, Liu X, Hu J, Zhu J. Empirical studies on the NLP techniques for source code data preprocessing [A]. Proceedings of the 2014 3rd International Workshop on Evidential Assessment of Software Technologies (EAST 2014) [C]. USA: ACM, 2014. 32-39.
- [10] 胡芳槐. 基于多种数据源的中文知识图谱构建方法研究[D]. 华东理工大学, 2015.
HU Fang-huai. Chinese Knowledge Graph Construction Method Based on Multiple Data Sources [D]. East China University of Science and Technology, 2015. (in Chinese)
- [11] Sarrafzadeh B, Vechtomova O, Jokic V. Exploring knowledge graphs for exploratory search [A]. Proceedings of the 5th Information Interaction in Context Symposium [C]. USA: ACM, 2014. 135-144.
- [12] Sarrafzadeh B, Vechtomova O. Combining document retrieval with knowledge graphs for exploratory search [A]. Proceedings of the 5th Information Interaction in Context Symposium [C]. USA: ACM, 2014. 345-347.
- [13] Lv Q, Xu L, Yu J, et al. Research on domain knowledge graph based on the large scale online knowledge fragment [A]. Proceedings of IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA) [C]. USA: IEEE, 2014. 312-315.
- [14] Li P, Wsng B, Shi Z, et al. Tag-textrank: a webpage keyword extraction method based on tags [J]. Journal of Computer Research and Development, 2012, 49(11): 2344-2351.
- [15] Tanon T P, Vrandečić D, Schaffert S, et al. From freebase to wikidata: the great migration [A]. Proceedings of the 25th International Conference on World Wide Web [C]. USA: ACM, 2016. 1419-1428.
- [16] Marchionini G. Exploratory search: from finding to understanding [J]. Communications of the ACM, 2006, 49(4): 41-46.
- [17] 钟秀琴, 刘忠, 丁盘苹. 基于混合推理的知识库的构建及其应用研究 [J]. 计算机学报, 2012, 35(4): 761-766.
ZHONG Xiu-qin, LIU Zhong, DING Pan-ping. Construction of knowledge base on hybrid reasoning and its application [J]. Journal of Computer Research and Development, 2012, 35(4): 761-766. (in Chinese)
- [18] 陈海鹏, 申铨京, 龙建武, 等. 自动确定聚类个数的模糊

- 聚类算法[J]. 电子学报, 2017, 45(3): 687 – 694.
- CHEN Hai-peng, SHEN Xuan-jing, LONG Jian-wu, et al. Fuzzy clustering algorithm for automatic identification of clusters[J]. Acta Electronica Sinica, 2017, 45(3): 687 – 694. (in Chinese)
- [19] Lin Z, Zhao J, Xie B. A graph database based crowdsourcing infrastructure for modelling and searching code structure[A]. Proceedings of the 6th Asia-Pacific Symposium on Internetwork[C]. USA: ACM, 2014. 15 – 24.
- [20] Su Y, Yang S, Sun H, et al. Exploiting relevance feedback in knowledge graph search[A]. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. USA: ACM, 2015. 1135 – 1144.
- [21] Zhu Y, Cao X, Bian Y, et al. CKGHV: A comprehensive knowledge graph for history visualization[A]. Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries[C]. USA: IEEE, 2014. 437 – 438.
- [22] Golovchinsky G, Adcock J, Pickens J, et al. Cerchiamo: A collaborative exploratory search tool[A]. Proceedings of Computer Supported Cooperative Work (CSCW)[C]. USA: ACM, 2008. 8 – 12.
- [23] O'Connor B, Krieger M, Ahn D. TweetMotif: Exploratory search and topic summarization for twitter[A]. Proceedings of the Fourth International Conference on Weblogs and Social Media (ICWSM)[C]. USA: AAAI, 2010. 1 – 2.
- [24] Lee J, Fuxman A, Zhao B, et al. Leveraging knowledge bases for contextual entity exploration[A]. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. USA: ACM, 2015. 1949 – 1958.
- [25] 林泽琦, 赵俊峰, 谢冰. 一种基于图数据库的代码结构解析与搜索方法[J]. 计算机研究与发展, 53(3): 531 – 540.
- LIN Ze-qi, ZHAO Jun-feng, XIE Bing. A graph database based method for parsing and searching code structure[J]. Journal of Computer Research and Development, 53(3): 531 – 540. (in Chinese)

作者简介



孙小兵(通信作者) 男. 1985年出生, 江苏姜堰人, 博士、副教授、硕士生导师, 主要研究领域为软件分析、维护与演化.
E-mail: xbsun@yzu.edu.cn



王璐 女. 1992年出生, 江苏镇江人, 硕士研究生, 研究方向为 bug 分析与修复.
E-mail: 1504385791@qq.com