任务指派是软件开发过程中最常见的一类开发管理活动,对提高软件开发效率和质量具有非常重要的意义.针对 issue 指派问题,Anvic 等人[24]首先利用机器学习算法,基于文本信息找到与给定 issue 类似的历史 issue,然后将参与这些历史 issue 修订的开发者进行排序,并指派给指定 issue.Jeong 和 Bhattacharya 等人[25,26]则基于给定 issue 的演化历史构造一个图,然后基于此图,利用分类方法来寻找合适的解决人员.Yu 等人研究了 pull-requests 的过载问题,提出了基于开发者社交信息和技术兴趣的评阅人推荐技术.

为查找合适的合作者,Surian 等人[27]构造了名为"开发项目语言/软件类"的网络,基于这个网络,定义了开发者社交关系距离的度量方法,并基于该距离来推荐潜在的合作者.Canfora 等人[28]则通过分析邮件列表和版本控制来了解开发者的社交和技术活跃度,从而为新加入的参与者推荐更好的"引路人".

这些技术专家推荐的相关研究主要探讨了如何从开发者的历史开发活动和社交行为来做推荐,这些研究为本文如何度量开发者的技术能力和技术兴趣提供了依据.

4.3 知识文档推荐

不同于商业软件,开源软件开发过程通常缺乏规范的文档,这给开发人员对这些软件的复用带来了困难,但是互联网上的丰富文档资料成为这些软件非常重要的知识来源.例如在 Stackoverflow 中,关于 Android API 的讨论文档覆盖了 Android 所有类文件的 87%,而且这些讨论被浏览了超过 7 千万次[29].当前,很多研究人员对面向开发者的知识文档推荐问题展开研究.

Favre 等人[30]提出了一种基于模式匹配的方法来定位源代码对应的文档.Chen 等人[31]则将正则表达式、关键字和聚类方法等结合起来,利用 SVM 模型提高文档推荐的准确度.此外,Rigby 等人[32]通过提取 posts 中的代码判断来帮助查找相关的讨论.

此外,Alberto 等人[33]设计了一个 Eclipse 插件,该插件能够根据开发者在编辑器中输入的代码来自动链接到相应的 StackOverflow 讨论帖.Tao 等人[34]聚焦于将项目 issue 与 Stackoverflow 相关讨论的自动联接,提出了一种将语义相似度和时间局部性相结合的方法,发现和定位与 issue 紧密相关的讨论贴,从而保证项目参与者获得 issue 更全面的信息,以更好地解决该 issue.

5 总 结

近年来,GitHub 社交化开源开发社区中,发布的开源项目和参与开源的开发者保持持续高速的增长.但是其中大量的开源项目仅仅是昙花一现,还有一些项目则仅仅只是吸引了很少量的开发者参与其中.如何实现开发者个人兴趣与开源项目技术需求之间的最佳匹配,是推动开源项目持续快速发展的关键.一方面,从项目的角度,吸引到最合适的开发者积极踊跃的参与和持续贡献是开源项目获得成功的关键;另一方面,从开发者的角度,每个开发者的精力是有限的,从海量开源项目中找到自己最感兴趣的项目并合理地分配时间持续参与,是开发者参与开源不断成长的重要因素.但是,开源资源库的大规模和高速增长以及开发者参与活动数据的高度分散,为实现这种开发者与开源项目的匹配带来巨大挑战.

本文提出了 RepoLike 的方法,充分利用 GitHub 中开源项目与开发者行为活动数据,综合多个维度的特征,基于 LTR 实现对开发者进行个性化开源项目推荐.具体的,我们综合考虑开源项目自身流行度、开源项目之间的技术相关度以及开发者之间的社交关联度这 3 个不同的维度的特征,通过对开发者已经参与项目的深度对开源项目进行排序并构建训练数据集,利用 LTR 的方法构建基于 3 个维度信息的开发者个性化开源项目推荐模型.最后,本文对所提方法进行了全面深入的实验验证,对比分析了基于线性综合的推荐方法与基于 LTR 的推荐模型在考虑不同维度信息以及不同时间间隔对实验结果的影响.实验结果表明:本文所提方法综合考虑多维信息,能够以较高的准确度实现面向开发者的个性化开源项目推荐.

在未来,我们可以从不同的方面做进一步的工作:首先,现在开源项目之间的技术依赖关系我们只考虑了评论里的技术参数,这在很大程度上可能会丢失那些没有出现在评论里的真正的技术依赖关系,我们将溯源源代码,挖掘出更深层次的依赖关系;其次,在 GitHub 中,开发者之间除了在 issue 和 pull-request 中的直接交互外,还有如共同修改过相同的代码文件等间接交互信息,下一步我们将进一步挖掘开发者之间的社交关联;最后,为了

更精确地评估本文方法的效果,我们会将推荐结果发送给部分 GitHub 中相应的开发人员,获取他们对推荐结果的反馈,从而获得更准确的关于推荐效果的评价.

致谢 本文实验和撰写过程中得到国防科学技术大学余跃博士的指导和建议,在此表示感谢.

References:

[1] Boyd DM, Ellison NB. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 2007, 13 (1):210−230. [doi: 10.1111/j.1083-6101.2007.00393.x]

[2] Storey MA, Treude C, van Deursen A, Cheng LT. The impact of social media on software engineering practices and tools. In: Proc. of the FSE/SDP Workshop on Future of Software Engineering Research. 2010. 359−364. [doi: 10.1145/1882362.1882435]

[3] Begel A, DeLine R, Zimmermann T. Social media for software engineering. In: Proc. of the FSE/SDP Workshop on Future of Software Engineering Research. 2010. 33−38.

[4] Begel A, Bosch J, Storey MA. Social networking meets software development: Perspectives from GitHub, MSDN, stack exchange, and TopCoder. IEEE Software, 2013,30(1):52−66. [doi: 10.1109/MS.2013.13]

[5] Dabbish L, Stuart C, Tsay J, Herbsleb J. Social coding in GitHub: Transparency and collaboration in an open software repository. In: Proc. of the CSCW. 2012. 1277−1286.

[6] Yu Y, Wang HM, Yin G, Wang T. Reviewer recommendation for pull-requests in GitHub: What can we learn from code review and bug assignment? Information and Software Technology Journal, 2016,74:204−218. [doi: 10.1016/j.infsof.2016.01.004]

[7] Wang HM, Yin G, Xie B, Liu XD, Wei J, Liu JN. Research on network-based large-scale collaborative development and evolution of trustworthy software. Scientia Sinica Informationis, 2014,44(1):1−19 (in Chinese with English abstract).

[8] Zhou M, Mockus A. Does the initial environment impact the future of developers? In: Proc. of the 33rd Int'l Conf. on Software Engineering. ACM Press, 2011. 271−280. [doi: 10.1145/1985793.1985831]

[9] Blincoe K, Harrison F, Damian D. Ecosystems in GitHub and a method for ecosystem idendification using reference coupling. In: Proc. of the Mining Software Repositories (MSR). 2015. [doi: 10.1109/MSR.2015.26]

[10] Zhu J, Shen B, Hu F. A learning to rank framework for developer recommendation in software crowdsourcing. In: Proc. of the 2015 Asia-Pacific Software Engineering Conf. (APSEC). IEEE, 2015. 285−292. [doi: 10.1109/APSEC.2015.50]

[11] Chen X, Zhang Y, Xu T, Qin Z. Learning to rank features for recommendation over multiple categories. In: Proc. of the 39th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM Press, 2016. 305−314. [doi: 10.1145/2911451.2911549]

[12] Yang C, Fan Q, Wang T, Wang HM. RepoLike: Personal repositories recommendation in social coding communities. In: Proc. of the 8th Asia-Pacific Symp. on Internetware on Internetware. ACM Press, 2016. [doi: 10.1145/2993717.2993725]

[13] Ye Y, Fischer G. Information delivery in support of learning reusable software components on demand. In: Proc. of the 2002 Int'l Conf. on Intelligent User Interfaces (IUI 2002). 2002. 159−166. [doi: 10.1145/502716.502741]

[14] Chen MG, Fu C, Xie Q, McMillan C, Poshyvanyk D, Cumby C. A search engine for finding highly relevant applications. In: Proc. of 2010 ACM/IEEE the 32nd Int'l Conf. on Software Engineering. 2010. 475−484. [doi: 10.1145/1806799.1806868]

[15] McMillan C, Poshyvanyk D, Grechanik M. Recommending source code examples via API call usages and documentation. In: Proc. of the 2nd Int'l Workshop on Recommendation Systems for Software Engineering (RSSE 2010). 2010. [doi: 10.1145/1808920.1808925]

[16] Lozano A, Kellens A, Mens K. Mendel: Source code recommendation based on a genetic metaphor. In: Proc. of the 26th IEEE/ACM Int'l Conf. on Automated Software Engineering. IEEE Computer Society, 2011. 384−387. [doi: 10.1109/ASE.2011. 6100078]

[17] Holmes R, Murphy GC. Using structural context to recommend source code examples. In: Proc. of the ICSE 2005. 2005. 117−125. [doi: 10.1145/1062455.1062491]

[18] Xie T, Pei J. MAPO: Mining API usages from open source repositories. In: Proc. of the 2006 Int'l Workshop on Mining Software Repositories. 2006. 54−57. [doi: 10.1145/1137983.1137997]

[19] Zagalsky A, Barzilay O, Yehudai A. Example overflow: Using social media for code recommendation. In: Proc. of the 3rd Int'l Workshop on Recommendation Systems for Software Engineering. IEEE Press, 2012. 38−42. [doi: 10.1109/RSSE.2012.6233407]

[20] Bajracharya S, Ossher J, Lopes C. Sourcerer: An internet-scale software repository. In: Proc. of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation. 2009. 1−4. [doi: 10.1109/SUITE.2009.5070010]

[21] Kokkoras F, Ntonas K, Kritikos A, Kakarontzas G, Stamelos I. Federated search for open source software reuse. In: Proc. of the 38th EUROMICRO Conf. on Software Engineering and Advanced Applications (SEAA). 2012. 200−203. [doi: 10.1109/SEAA.2012.55]

[22] Yin G, Wang T, Wang HM, Fan Q, Zhang Y, Yu Y, Yang C. OSSEAN: Mining crowd wisdom in open source communities. In: Proc. of the 2015 IEEE Symp. on Service-Oriented System Engineering (SOSE). IEEE, 2015. 367−371. [doi: 10.1109/SOSE.2015.51]

[23] Brandt J, Dontcheva M, Weskamp M, Klemmer SR. Example-Centric programming: Integrating web search into the development environment. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. 2010. 513−522. [doi: 10.1145/1753326.1753402]

[24] Anvik J, Hiew L, Murphy GC. Who should fix this bug? In: Proc. of the ICSE. 2006. 361−370. [doi: 10.1145/1134285.1134336]

[25] Bhattacharya P, Neamtiu I. Fine-Grained incremental learning and multi-feature tossing graphs to improve bug triaging. In: Proc. of the ICSM. 2010. 1−10. [doi: 10.1109/ICSM.2010.5609736]

[26] Jeong G, Kim S, Zimmermann T. Improving bug triage with bug tossing graphs. In: Proc. of the FSE. 2009. 111−120. [doi: 10.1145/1595696.1595715]

[27] Surian D, Liu N, Lo D, Tong HH, Lim EP, Faloutsos C. Recommending people in developers' collaboration network. In: Proc. of the WCRE. 2011. 379−388. [doi: 10.1109/WCRE.2011.53]

[28] Canfora G, Di Penta M, Oliveto R, Panichella S. Who is going to mentor newcomers in open source projects? In: Proc. of the FSE. 2012. 44. [doi: 10.1145/2393596.2393647]

[29] Allamanis M, Sutton C. Why, when, and what: Analyzing stack overflow questions by topic, type, and code. In: Proc. of the MSR. 2013. 53−56. [doi: 10.1109/MSR.2013.6624004]

[30] Favre JM, Lammel R, Leinberger M, Schmorleiz T, Varanovich A. Linking documentation and source code in a software chrestomathy. In: Proc. of the 19th Working Conf. on Reverse Engineering (WCRE). 2012. 335−344. [doi: 10.1109/WCRE.2012.43]

[31] Chen X, Grundy J. Improving automated documentation to code traceability by combining retrieval techniques. In: Proc. of the 26th IEEE/ACM Int'l Conf. on Automated Software Engineering. 2011. 223−232. [doi: 10.1109/ASE.2011.6100057]

[32] Rigby PC, Robillard MP. Discovering essential code elements in informal documentation. In: Proc. of the 2013 Int'l Conf. on Software Engineering. 2013. 832−841. [doi: 10.1109/ICSE.2013.6606629]

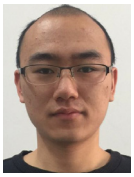
[33] Bacchelli A, Ponzanelli L, Lanza M. Harnessing stack overflow for the IDE. In: Proc. of the 3rd Int'l Workshop on Recommendation Systems for Software Engineering (RSSE). 2012. 26−30. [doi: 10.1109/RSSE.2012.6233404]

[34] Wang T, Yin G, Wang HM, Yang C, Zou P. Linking stack overflow to issue tracker for issue resolution. In: Proc. of the 6th Asia-Pacific Symp. on Internetware on Internetware. ACM Press, 2014. 11−14. [doi: 10.1145/2677832.2677839]

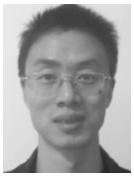
附中文参考文献:

[7] 王怀民,尹刚,谢冰,刘旭东,魏峻,刘江宁.基于网络的可信软件大规模协同开发与演化.中国科学:信息科学,2014,44(1):1−19.

杨程(1991－),男,湖南桃江人,学士,主要研究领域为开源软件工程,数据挖掘,开源软件中的推荐.



范强(1990－),男,硕士,CCF 专业会员,主要研究领域为开源软件工程,数据挖掘.



王涛(1984－),男,博士,助理研究员,主要研究领域为开源软件工程,机器学习,数据挖掘,开源软件中的知识发现.

尹刚(1975－),男,博士,副教授,CCF 专业会员,主要研究领域为分布式计算,信息安全,软件工程,机器学习.

王怀民(1962－),男,博士,教授,博士生导师, CCF 会士,主要研究领域为中间件,软件代理,可信计算.