后, 我们通过语义一致性变换实验, 验证了数据泄露问题影响了神经程序修复系统的鲁棒性; 最后, 我们提出了一个数据收集、过滤和划分策略, 并在此基础上构建了一个纯净的数据集, 并在 Clean4J_Benchmark 数据集上重新训练了神经程序修复模型, 验证了我们制定的数据收集、过滤和划分策略的有效性.

数据泄露问题对于神经程序修复系统的重要性不言而喻, 因此有必要对神经程序修复系统中的数据泄露问题继续进行研究. 后续研究可以从以下 3 个方面开展: 首先, 本文并未对基于大语言模型(参数量超过 10 亿)的神经程序修复系统进行研究, 由于大语言模型的训练数据十分庞大, 对于其数据泄露的定义以及检测都具有挑战性, 研究人员可以在这个场景下对神经程序修复系统的数据泄露问题继续研究, 并且设计更加完备的过滤策略以及收集更多数据, 构建一个更为大型的神经程序修复系统数据集, 供基于大语言模型的神经程序修复系统进行训练或微调; 其次, 研究人员可以对神经程序修复系统常用的基准测试集进行系统的实证研究, 并在此基础上设计一个与现有的神经程序修复系统训练数据没有重叠的基准测试, 保证该基准测试的纯净, 这样可以避免数据泄露、保证在该基准测试上进行评估的公平性; 最后, 研究人员可以继续深入研究数据泄露会对神经程序修复系统造成哪些负面影响, 并设法消除这些影响.

References:

[1] Monperrus M. Automatic software repair: A bibliography. ACM Computing Surveys, 2018, 51(1): 1−24.

[2] Goues CL, Pradel M, Roychoudhury A. Automated program repair. Communications of the ACM, 2019, 62(12): 56−65. [doi: 10.1145/3318162]

[3] Zhang Q, Fang C, Ma Y, *et al*. A survey of learning-based automated program repair. arXiv:2301.03270, 2023.

[4] Le Goues N, Le Goues C, Nguyen T, Forrest S, Weimer W. GenProg: A generic method for automatic software repair. IEEE Trans. on Software Engineering, 2012, 38(1): 54−72. [doi: 10.1109/TSE.2011.104]

[5] Martinez M, Monperrus M. ASTOR: A program repair library for Java. In: Proc. of the 25th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2016. 441−444. [doi: 10.1145/2931037.2948705]

[6] Yuan Y, Banzhaf W. ARJA: Automated repair of Java programs via multi-objective genetic programming. IEEE Trans. on Software Engineering, 2018, 46(10): 1040−1067. [doi: 10.1109/TSE.2018.2874648]

[7] Jiang JJ, Xiong YF, Zhang HY, Gao Q, Chen XQ. Shaping program repair space with existing patches and similar code. In: Proc. of the 27th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2018. 298−309.

[8] Durieux T, Monperrus M. DynaMoth: Dynamic code synthesis for automatic program repair. In: Proc. of the 11th Int'l Workshop on Automation of Software Test. 2016. 85−91. [doi: 10.1145/2896921.2896931]

[9] Xuan J, Martinez M, Demarco F, *et al*. Nopol: Automatic repair of conditional statement bugs in Java programs. IEEE Trans. on Software Engineering, 2016, 43(1): 34−55. [doi: 10.1109/TSE.2016.2560811]

[10] Martinez M, Monperrus M. Ultra-large repair search space with automatically mined templates: The cardumen mode of Astor. In: Proc. of the Int'l Symp. on Search Based Software Engineering (SSBSE). Springer, 2018. 65−86.

[11] Mechtaev S, Yi J, Roychoudhury A. Angelix: Scalable multiline program patch synthesis via symbolic analysis. In: Proc. of the 38th Int'l Conf. on Software Engineering (ICSE). IEEE, 2016. 691−701. [doi: 10.1145/2884781.2884807]

[12] Liu K, Koyuncu A, Kim D, *et al*. AVATAR: Fixing semantic bugs with fix patterns of static analysis violations. In: Proc. of the 26th IEEE Int'l Conf. on Software Analysis, Evolution and Reengineering (SANER). IEEE, 2019. 1−12.

[13] Liu K, Koyuncu A, Kim D, *et al*. TBar: Revisiting template-based automated program repair. In: Proc. of the 28th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2019. 31−42. [doi: 10.1145/3293882.3330577]

[14] Kim D, Nam J, Song J, Kim S. Automatic patch generation learned from human-written patches. In: Proc. of the 35th Int'l Conf. on Software Engineering (ICSE). IEEE, 2013. 802−811. [doi: 10.1109/ICSE.2013.6606626]

[15] Hua JR, Zhang MS, Wang KY, Khurshid S. Towards practical program repair with on-demand candidate generation. In: Proc. of the 40th Int'l Conf. on Software Engineering (ICSE). ACM, 2018. 12−23. [doi: 10.1145/3180155.3180245]

[16] Huang K, Xu Z, Yang S, *et al*. A survey on automated program repair techniques. arXiv:2303.18184, 2023.

[17] Samala RK, Chan HP, Hadjiiski L, *et al*. Hazards of data leakage in machine learning: A study on classification of breast cancer using deep neural networks. In: Proc. of the Medical Imaging 2020: Computer-aided Diagnosis, Vol.11314. SPIE, 2020. 279−284. [doi: 10.1117/12.2549313]

[18] Tufano M, Pantiuchina J, Watson C, *et al*. On learning meaningful code changes via neural machine translation. In: Proc. of the 41st Int'l Conf. on Software Engineering (ICSE). IEEE, 2019. 25–36. [doi: 10.1109/ICSE.2019.00021]

[19] Tufano M, Watson C, Bavota G, *et al*. An empirical study on learning bug-fixing patches in the wild via neural machine translation. ACM Trans. on Software Engineering and Methodology, 2019, 28(4): 1–29. [doi: 10.1145/3340544]

[20] Jin M, Shahriar S, Tufano M, *et al*. InferFix: End-to-end program repair with llms. arXiv:2303.07263, 2023.

[21] Xia CS, Ding Y, Zhang L. Revisiting the plastic surgery hypothesis via large language models. arXiv:2303.10494, 2023.

[22] Xia CS, Zhang L. Less training, more repairing please: Revisiting automated program repair via zero-shot learning. In: Proc. of the 2022 30th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/FSE). ACM, 2022. 959–971. [doi: 10.1145/3540250.3549101]

[23] Chen X, Lin M, Schärli N, *et al*. Teaching large language models to self-debug. arXiv:2304.05128, 2023.

[24] Xia CS, Wei Y, Zhang L. Automated program repair in the era of large pre-trained language models. In: Proc. of the 45th Int'l Conf. on Software Engineering (ICSE). IEEE, 2023.

[25] Cao J, Li M, Wen M, *et al*. A study on prompt design, advantages and limitations of ChatGPT for deep learning program repair. arXiv:2304.08191, 2023.

[26] Xia CS, Zhang L. Conversational automated program repair. arXiv:2301.13246, 2023.

[27] Bellon S, Koschke R, Antoniol G, *et al*. Comparison and evaluation of clone detection tools. IEEE Trans. on Software Engineering, 2007, 33(9): 577–591. [doi: 10.1109/TSE.2007.70725]

[28] Ji SL, Du TY, Deng SG, *et al*. Robustness certification research on deep learning models: A survey. Chinese Journal of Computers, 2022, 45(1): 190–206 (in Chinese with English abstract).

[29] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014.

[30] Cheng Y, Tu ZP, Meng FD, *et al*. Towards robust neural machine translation. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Vol. 1. ACL, 2018. 1756–1766. [doi: 10.18653/v1/P18-1163]

[31] Ge H, Zhong W, Li C, *et al*. RobustNPR: Evaluating the robustness of neural program repair models. Journal of Software: Evolution and Process, 2024, 36(4): e2586. [doi: 10.1002/smr.2586]

[32] Goyal C. Data leakage and its effect on the performance of an ML model. 2021. https://www.analyticsvidhya.com/blog/2021/07/data-leakage-and-its-effect-on-the-performance-of-an-ml-model/

[33] Elangovan A, He J, Verspoor K. Memorization vs. generalization: Quantifying data leakage in NLP performance evaluation. arXiv:2102.01818, 2021.

[34] Lewis P, Stenetorp P, Riedel S. Question and answer test-train overlap in open-domain question answering datasets. arXiv:2008.02637, 2020.

[35] Jiang N, Liu K, Lutellier T, *et al*. Impact of code language models on automated program repair. arXiv:2302.05020, 2023.

[36] Zhu Q, Sun Z, Xiao Y, *et al*. A syntax-guided edit decoder for neural program repair. In: Proc. of the 2021 29th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering (ESEC/FSE). ACM, 2021. 341–353.

[37] Chen ZM, Kommrusch SJ, Tufano M, Pouchet LN, Poshyvanyk D, Monperrus M. Sequencer: Sequence-to-sequence learning for end-to-end program repair. IEEE Trans. on Software Engineering, 2019, 47(9): 1943–1959. [doi: 10.1109/TSE.2019.2940179]

[38] Ye H, Martinez M, Monperrus M. Neural program repair with execution-based backpropagation. In: Proc. of the 44th Int'l Conf. on Software Engineering (ICSE). IEEE, 2022. 1506–1518. [doi: 10.1145/3510003.3510222]

[39] Meng X, Wang X, Zhang H, *et al*. Improving fault localization and program repair with deep semantic features and transferred knowledge. In: Proc. of the 44th Int'l Conf. on Software Engineering (ICSE). IEEE, 2022. 1169–1180.

[40] Lutellier T, Pham HV, Pang L, *et al*. Coconut: Combining context-aware neural translation models using ensemble for program repair. In: Proc. of the 29th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2020. 101–114.

[41] Jiang N, Lutellier T, Tan L. Cure: Code-aware neural machine translation for automatic program repair. In: Proc. of the 43rd Int'l Conf. on Software Engineering (ICSE). IEEE, 2021. 1161–1173. [doi: 10.1109/ICSE43902.2021.00107]

[42] Li Y, Wang S, Nguyen TN. DLFix: Context-based code transformation learning for automated program repair. In: Proc. of the 42nd Int'l Conf. on Software Engineering (ICSE). IEEE, 2020. 602–614. [doi: 10.1145/3377811.3380345]

[43] Li Y, Wang S, Nguyen TN. DEAR: A novel deep learning-based approach for automated program repair. In: Proc. of the 44th Int'l Conf. on Software Engineering (ICSE). IEEE, 2022. 511−523. [doi: 10.1145/3510003.3510177]

[44] Yuan W, Zhang Q, He T, *et al*. CIRCLE: Continual repair across programming languages. In: Proc. of the 31st ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2022. 678−690. [doi: 10.1145/3533767.3534219]

[45] Feng Z, Guo D, Tang D, *et al*. CodeBERT: A pre-trained model for programming and natural languages. arXiv:2002.08155, 2020.

[46] Raffel C, Shazeer N, Roberts A, *et al*. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 2020, 21(1): 5485−5551. [doi: 10.5555/3455716.3455856]

[47] Saha RK, Lyu Y, Lam W, *et al*. Bugs.jar: A large-scale, diverse dataset of real-world java bugs. In: Proc. of the 15th Int'l Conf. on Mining Software Repositories (MSR). IEEE, 2018. 10−13. [doi: 10.1145/3196398.3196473]

[48] Just R, Jalali D, Ernst MD. Defects4J: A database of existing faults to enable controlled testing studies for Java programs. In: Proc. of the 23rd ACM SIGSOFT Int'l Symp. on Software Testing and Analysis (ISSTA). ACM, 2014. 437−440.

[49] Durieux T, Monperrus M. IntroClassJava: A benchmark of 297 small and buggy Java programs. 2016. https://hal.science/hal-01272126/

[50] Lin D, Koppel J, Chen A, *et al*. QuixBugs: A multi-lingual program repair benchmark set based on the Quixey challenge. In: Proc. Companion of the 2017 ACM SIGPLAN Int'l Conf. on Systems, Programming, Languages, and Applications: Software for Humanity. ACM, 2017. 55−56.

[51] Lawler R. How do you hire great engineers? Give them a challenge. 2012. https://www.eejournal.com/fresh_bytes/how-do-you-hire-great-engineers-give-them-a-challenge/

[52] Zhong W, Ge H, Ai H, *et al*. StandUp4NPR: Standardizing setup for empirically comparing neural program repair systems. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering (ASE). IEEE, 2022. 1−13.

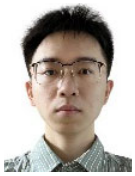
[53] Jiang JJ, Chen JJ, Xiong YF. Survey of automatic program repair techniques. Ruan Jian Xue Bao/Journal of Software, 2021, 32(9): 2665−2690 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6274.htm [doi: 10.13328/j.cnki.jos.006274]

附中文参考文献:

[28] 纪守领, 杜天宇, 邓水光, 等. 深度学习模型鲁棒性研究综述. 计算机学报, 2022, 45(1): 190−206.

[53] 姜佳君, 陈俊洁, 熊英飞. 软件缺陷自动修复技术综述. 软件学报, 2021, 32(9): 2665−2690. http://www.jos.org.cn/1000-9825/6274.htm [doi: 10.13328/j.cnki.jos.006274]

李卿源(2000－), 男, 硕士生, CCF 学生会员, 主要研究领域为软件工程, 自然语言处理.



钟文康(1997－), 男, 博士生, 主要研究领域为软件工程, 自然语言处理, 程序自动修复.

李传艺(1991－), 男, 博士, 准聘助理教授, 博士生导师, CCF 专业会员, 主要研究领域为软件工程, 业务过程管理, 自然语言处理.

葛季栋(1978－), 男, 博士, 副教授, 博士生导师, CCF 高级会员, 主要研究领域为自然语言处理, 智能软件工程, 分布式计算, 边缘计算, 服务计算, 业务过程管理.

骆斌(1967－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为分布式计算, 边缘计算, 自然语言处理, 智能软件工程.