Sen Fang

☎ (+1) 984-379-1827 • ✉ sfang9@ncsu.edu • 🌐 tomasandersonfang.github.io

Research Interests

I am passionate about advancing research at the cutting edge of large language models (LLMs) and automated software engineering. Recently, I have focused on leveraging LLMs to automate key software engineering tasks, including program repair (using fine-tuning with tailored code representations and LoRA), code optimization (fine-tuning with diff-based output), test data generation (multi-level prompting), and bug-report-driven maintenance (pre-training and fine-tuning). My research spans three main areas: 1) exploring how code representations influence LLM performance in software engineering; 2) utilizing AI agents for end-to-end automated software development; and 3) investigating the potential of LLMs for AI-centric smart contract development.

Education

NC State University — **2024-2029 (Expected)**
Ph.D. in Computer Science, GPA:- — *Advisor: To be decided*

Central China Normal University — **2018-2020**
MSc in Electronics and Communication Engineering, GPA:3.44/5.0 — *Advisor: Shaocheng Qu*
Thesis: Research on Machine Translation Model Based on Self-Attention Mechanism

Wuhan Polytechnic University — **2014-2018**
BSc in Electronic Information Engineering, GPA: 3.47/5.0 — *Advisor: Lei Yang*
Thesis: Performance Comparison of Direct Sequence Spread Spectrum Systems under Different Modulation Methods

Outstanding Student 2018.

Professional Experience

KTH Royal Institute of Technology — **2023.03-2024.03**
Research Engineer — *Advisor: Martin Monperrus*

Macau University of Science and Technology — **2020.09-2022.11**
Others — *Advisor: Tao Zhang*

Research Experience

Completed Research

Repairllama: Efficient representations and fine-tuned adapters for program repair — **2023.09–2024.12**
https://github.com/ASSERT-KTH/repairllama — *GitHub*
In this work, we explore the impact of various code representations on program repair. Then, we use our best-designed representation to build an LLM-based program repair model, RepairLLaMA.

Generative AI to Generate Test Data Generators — **2023.06–2024.02**
https://github.com/ASSERT-KTH/lollm — *GitHub*
In this work, we explore to what extent an LLM, i.e., ChatGPT, can generate test data generators that can keep consistent with the different cultural backgrounds.

Supersonic: Learning to Generate Source Code Optimisations in C/C++ — **2023.05–2024.04**
https://github.com/ASSERT-KTH/Supersonic — *GitHub*
We propose Supersonic, an LLM-based approach for source code-level code optimization, which outperforms GPT-3.5 and GPT-4 in our comparative experiments.

RepresentThemAll: A Universal Learning Representation of Bug Reports — **2021.03–2022.12**
https://github.com/ICSE-2023/RepresentThemALL — *GitHub*

Automated Pull Request Title Generation — **2020.11–2021.09**
https://github.com/TomasAndersonFang/PRHAN — *GitHub*

Bug Priority Prediction — **2020.09–2021.05**
https://github.com/TanYoushuai123/PPWGCN — *GitHub*

Neural Code Search — **2020.06–2021.02**
https://github.com/TomasAndersonFang/SANCS — *GitHub*

Ongoing Research

Deep Smart Contract Intent Detection **2022.07–Now**
https://gitlab.com/web3se/smartintent *GitLab*

- Smart contracts are often accompanied by huge economic effects, so their security is a very important issue. We try to define a new task: automated smart contract intent detection. This task can automatically detect risky intent in a smart contract.
- We design an automated model SMARTINTENTNN, which is a highlight approach based on K-means and Bi-LSTM.
- We collected the first dataset used for smart contract intent detection.

Publications

Published

- **Fang, Sen**, Tao Zhang, Youshuai Tan, He Jiang, Xin Xia, and Xiaobing Sun. "RepresentThemAll: A Universal Learning Representation of Bug Reports." In Proceedings of the 45th International Conference on Software Engineering. 2023.
- Yuan, Dawei*, **Fang, Sen***, Tao Zhang, Zhou Xu, and Xiapu Luo. "Java Code Clone Detection by Exploiting Semantic and Syntax Information From Intermediate Code-Based Graph." IEEE Transactions on Reliability (2022). (∗: Equal contribution)
- **Fang, Sen**, Tao Zhang, You-Shuai Tan, Zhou Xu, Zhi-Xin Yuan, and Ling-Ze Meng. "PRHAN: Automated Pull Request Description Generation Based on Hybrid Attention Network." Journal of Systems and Software 185 (2022): 111160.
- **Fang, Sen***, You-shuai Tan*, Tao Zhang, Zhou Xu, and Hui Liu. "Effective prediction of bug-fixing priority via weighted graph convolutional networks." IEEE Transactions on Reliability 70, no. 2 (2021): 563-574. (∗: Equal contribution)
- **Fang, Sen**, You-Shuai Tan, Tao Zhang, and Yepang Liu. "Self-attention networks for code search." Information and Software Technology 134 (2021): 106542.
- Chen, Zimin, **Fang, Sen**, and Martin Monperrus. "Supersonic: Learning to generate source code optimizations in C/C++." IEEE Transactions on Software Engineering (2024).
- Tan, Youshuai; Chen, Jinfu; Shang, Weiyi; Zhang, Tao; **Fang, Sen**; Luo, Xiapu; Chen, Zijie; Qi, Shuhao. "STRE: An Automated Approach to Suggesting App Developers When to Stop Reading Reviews." IEEE Transactions on Software Engineering.
- Li, Yao; Zhang, Tao; Luo, Xiapu; Cai, Haipeng; **Fang, Sen**; Yuan, Dawei. "Do Pre-trained Language Models Indeed Understand Software Engineering Tasks?" IEEE Transactions on Software Engineering.
- Huang, Youwei, Tao Zhang, **Fang, Sen**, and Youshuai Tan. "Deep Smart Contract Intent Detection." arXiv preprint arXiv:2211.10724 (2022).
- Huang, Youwei, Tao Zhang, **Fang, Sen**, and Youshuai Tan. "SmartIntentNN: Towards Smart Contract Intent Detection." arXiv preprint arXiv:2211.13670 (2022).
- Benoit Baudry; Khashayar Etemadi; **Fang, Sen**, etc. "Generative AI to Generate Test Data Generators" (IEEE Software). (Sort by alphabet)

Submitting

- André Silva*; **Fang, Sen***; Martin Monperrus. "RepairLLaMA: Efficient Representations and Fine-Tuned Adapters for Program Repair". (Submitted to TSE/Major Revision) (∗: Equal contribution)
- Yao Li; **Fang, Sen**, etc. "Enhancing Android Malware Detection: The Influence of ChatGPT on Decision-centric Tasks" (Submitted to TOSEM).

Technical skills

Programming Languages	Python, C/C++, Java, LaTeX
Frameworks	PyTorch, transformers, TensorFlow, JAX, NumPy, SLURM
Operating Systems	Unix/Linux, Windows
Development Environments	Linux Toolchain, Jupyter, PyCharm, Visual Studio Code