“JW” Jie Wu
Research Statement
October 6, 2023

Summary

We are entering a future where teams of intelligent bots will play a critical role in the software development process. Having worked in both interdisciplinary research topics and industrial software developments, I observe that a proper blend of recent advancements in AI, particularly *Large Language Models (LLMs)*, will unlock many exciting future directions in software engineering (SE). My research is driven by a strong desire to **systematically simplify** the complicated, time-consuming, and sometimes tedious software development process. My ambition is to create **automated, systematic, explainable decision systems** that can greatly ease the job of 27 million professional software engineers in the world. Ultimately, the software development process will be transformed into a series of decisions, supervised by engineers, who can then focus on more creative things. The outcome of my research will contribute to the exciting future of the transition from “Data-Driven Development” to “AI-Driven Development”.

Research Approach

I use a two-step approach to achieve the research goals. **First**, I focus on understanding and evaluating LLM for SE tasks such as code generation, in the context of software development. **Second**, I iteratively address any identified shortcomings or fundamental issues, which helps to ensure that LLM can be reliably and successfully applied to real-world SE tasks in practice.

Understanding and Benchmarking Large Language Model for Code

LLMs have significantly improved the ability to perform tasks in the field of code generation. However, there is still a gap between LLMs being capable coders and being top-tier software engineers. In my recent work [1, 2], based on the observation that top-level software engineers often ask clarifying questions to reduce ambiguity in both requirements and coding solutions, I argue that the same should be applied to LLMs for code generation tasks. I created a new benchmark for evaluating the degree of communication skills of LLMs for code. I conducted the first study on the degree of communication skills in code generation tasks of LLMs. My main finding is that 1) ChatGPT is currently **very weak** at asking clarifying questions when information is removed in problem description, 2) Lower temperature or using GPT-4 does not help much to increase the chance of LLM to ask questions, but the proposed *self-correcting LLM* is effective. This indicates that certain improvements in training data or model are needed for LLM of code to reduce the gap in communications.

Furthermore, besides the communication lens, I also plan to apply other different lens to better understand other fundamental aspects of LLM for code. This includes aspects such as documentation ability, logical reasoning, program analysis, context-awareness, fine-tuning efficiency, and cost. I view **explainability** as a top priority, because having a deeper understanding will open up new ideas to further improve LLM for code, and open up new application of LLM to other SE tasks.

Using Large Language Model in Building ML-Enabled Systems

As we are shifting from traditional software to ML-enabled systems to let computers automatically learn the parameters in programs, ML components are being added to more and more critical and impactful software systems. This is an emerging field, called *“AI Engineering”*, with a number of challenges on technical side such as the use of *A/B testing*, and collaboration side such as the use of *lifecycle models*.

During my PhD study, I proposed the first approach to formalize data-driven decisions from A/B testing results, using *Multi-Criteria Decision-Making (MCDM)* in a systematic way [3, 4]. With this improved formalization, I further attacked the problem of automating online A/B testing for data-intensive ML-enabled applications. I found that the existing single-objective methods for this problem are quite limited in industrial settings in practice, so I proposed a *Multi-Objective Evolutionary Algorithm (MOEA)* to address this issue [5, 6]. However, this approach still only automates the adjustment of parameter values in the system rather than creation of more sophisticated changes or features. To go further from "data-driven decisions" to "AI-driven decisions", I plan to leverage LLM to go further from optimizing local parameter values to automating more sophisticated changes, due to the remarkable capability of LLM in coding tasks.

On the other hand, my recent work [7] focused on the problem that traditional process models (such as waterfall, spiral or agile model) are limited in interdisciplinary collaboration when building products with ML components. I developed a set of propositions based on interviews with practitioners, and proposed Vee process model, *V4ML*, to address this issue. I found that V4ML process model requires more efforts on documentation, system decomposition, verification and validation (V&V), but it addressed the interdisciplinary collaboration challenges and additional complexity introduced by ML components. As future work, I plan to leverage LLM to build a **conversational tool**, to provide suggestions to developers guided by V4ML for improved engineering quality and reduced risks when building ML-enabled systems. I also have previous background and experience in building AI-based tools such as online sketch recognition [8, 9] and offline sketch parsing [10] for efficient flowchart creation. I will continue to identify and bridge the technical gaps when building such tools for SE tasks, with focuses on the efficiency, effectiveness and usability of the tools.

References

[1] Jie JW Wu. Benchmarking the communication competence in large language models for code generation. 2023. (Under Review).

[2] Jie JW Wu. Does asking clarifying questions increases confidence in generated code? on the communication skills of large language models. *In Proceedings of the 7th Annual Symposium on Machine Programming (MAPS '23)*, 2023.

[3] Jie JW Wu, Thomas A. Mazzuchi, and Shahram Sarkani. Comparison of multi-criteria decision-making methods for online controlled experiments in a launch decision-making framework. *Information and Software Technology*, 155:107–115, 2023.

[4] Jie JW Wu. *Towards Formalizing Data-Driven Decision-Making from Big Data: A Systematic Multi-Criteria Decision-Making Approach in Online Controlled Experiments*. PhD thesis, The George Washington University, 2023.

[5] Jie JW Wu, Thomas A. Mazzuchi, and Shahram Sarkani. A multi-objective evolutionary approach towards automated online controlled experiments. *Journal of Systems and Software*, 2023.

[6] Jie JW Wu. Can offline a/b testing be automated for data-driven requirement engineering? 2023. (Under Review).

[7] Jie JW Wu. Application of systems engineering process in ml-enabled systems. 2023. (Under Review).

[8] Jie Wu, Changhu Wang, Liqing Zhang, and Yong Rui. Sketch recognition with natural correction and editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[9] Jie Wu, Changhu Wang, Liqing Zhang, and Yong Rui. Smartvisio: Interactive sketch recognition with natural correction and editing. In *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014. (demo).

[10] Jie Wu, Changhu Wang, Liqing Zhang, and Yong Rui. Offline sketch parsing via shapeness estimation. In *IJCAI*, volume 15, 2015.