

ASE 2025 Trip Report

2025.11.17 ~ 2025.11.20, Seoul, Korea

고려대학교 소프트웨어분석연구실 김동욱



1 들어가며

올해는 서울에서 PL/SE 분야의 최상위 국제학술대회가 두 개나 개최되는 특별한 해이다. 저번 PLDI 2025에 이어서 이번 서울 워커히호텔에서 열린 ASE 2025에도 참가하게 되어 내가 들었던 인상깊은 발표들을 이 글을 통해 공유하려고 한다.

2 학회 발표들

Defects4C: Benchmarking Large Language Model Repair Capability with C/C++ Bugs

이 연구는 C/C++ 도메인의 프로그램 자동 수정을 위한 새로운 벤치마크를 제안한다. 정말 놀랍게도 이전에는 실제 세계의 C/C++ 언어를 대상으로 하는 “제대로 된” 버그 벤치마크가 없었다. 2년 전, 2023년에 BugsC++라는 C/C++ 버그 벤치마크가 공개되어 C/C++을 대상으로 FL 연구와 APR 연구가 가능해지기는 했다. 하지만 지금까지 했던 두 번의 연구 동안 BugsC++ 벤치마크를 적극적으로 사용하면서 솔직하게 완성도가 낮다고 생각하고 있었다 (중복 데이터, 유지 보수 없음 등의 이슈). 앞으로의 연구에 큰 도움이 될 잘 정립된 C/C++ 버그 벤치마크가 등장했다는 게 참 반가웠다.

이 연구에서 한 가지 주목해 볼만한 지점은 기존의 FL, APR 기술의 확장성이다. 실험 결과, 전통과 역사의 Java 언어 Defects4J 벤치마크와 다르게 LLM의 패치 성공률이 극히 낮다. 따라서, 기존의 FL, APR 연구들의 방법론을 일반적으로 확장할 수 있는 것인지, 아니면 새로운 접근방법이 필요한 것인지 확인하는 다양한 향후 연구들이 가능할 것 같다.

Efficient Understanding of Machine Learning Model Mispredictions

Machine Learning Model Diagnosis (MMD)라는 생소한 분야에 대한 연구인데, MMD는 머신러닝 모델의 잘못된 예측의 원인을 진단하는 기술이다. eXplainable AI와 다르게 모델이 어떤 특징들을 가진 입력에 대해서 실패하는 지에 대한 규칙(패턴)의 집합을 만든다. 구체적으로, 기존의 무거운 블랙박스 머신러닝 모델 대신에 의사결정트리를 사용하여 비용을 크게 절약하고 더 이해하기 쉬운 예측 실패 규칙을 찾는다. 기존의 블랙박스 모델을 투명한 접근방식으로 대체한다는 점에서 전민석 교수님이 연구했던 PL4XGL과도 비슷한데, 인공지능 모델의 explainability가 점점 주목받는 연구 방향이 될 것 같아서 소개한다.

Characterizing Multi-Hunk Patches: Divergence, Proximity, and LLM Repair Challenges

이 연구는 APR이 오랜 기간 동안 해결하지 못하고 있는 문제, multi-hunk bug에 대해 지적한다. APR이 고난도 기술이라서 기존 벤치마크들은 “패치가 성공할 만한” single-hunk bug로 구성하는 것이 목적인 반면에, 이 연구는 버그가 파일, 객체 등의 단위로 어떻게 분산되어 있는지를 분석하는 일종의 난이도 측정 작업을 통해 버그들을 분류한다.

perfect fault localization 환경에서도 최신 LLM 모델들이 여러 군데를 동시에 패치하는 것

은 성공률이 낮은 것으로 보아 APR은 여전히 갈 길이 먼 것 같다.

Let the Code Speak: Incorporating Program Dynamic State for Better Method-Level Fault Localization

결함 위치 추정 연구에서도 LLM을 사용하는 방식이 이제는 기본인 것처럼 자리잡았다. 하지만 LLM 기반 FL 기술들이 전통적인 FL 기술들과 다르게 실행 정보를 잘 활용하지 않는 모습에 코드만 보고 결함 위치를 찾는게 정말 효과가 좋은 것인가? 하는 의문이 들곤 했었다. 이 연구가 이런 나의 의문점을 정확하게 짚어주면서 흥미로운 해결 방법을 제시하고 있기 때문에 소개한다. 이 연구에서는 LLM과 동적 실행 정보를 결합하기 위해서 FL을 하는 에이전트(FL)와 디버깅 메시지를 삽입하는 에이전트(PD)를 두고 상호작용한다. 예를 들어, FL 에이전트가 특정 실행문이 의심스럽다고 지정하면 PD 에이전트가 그것과 관련된 디버깅 정보를 출력하는 실행문을 삽입하고 출력된 디버깅 정보를 다시 FL 에이전트가 활용하는 식이다. 두 개의 LLM 에이전트의 동작이 마치 실제 개발자의 디버깅 과정과 비슷하다고 느껴지는 흥미로운 접근 방법이었다.

3 마치며

지금까지 네 번의 국제학술대회 참석 기회가 있었지만 SE 학회에 참가하는 경험은 이번이 처음이었다. SE 학회는 PL 학회와 다르게 공연과 이벤트들을 준비해서 같이 즐기는 축제의 느낌이 조금 더 있는 것 같다. 또한, 이번 학회는 특이하게 발표시간이 10분으로 짧게 주어지고 저자와의 자세한 이야기는 포스터 세션에서 따로 진행하도록 했다. SE 학회도 출판되는 논문의 수가 점점 늘어나고 있기 때문에 학회의 모습이 앞으로 어떻게 바뀌게 될지도 궁금하다. 끝으로 학회 참석이라는 귀중한 경험을 할 수 있게 해주신 오학주 교수님께 감사드립니다.