



FSE 2026 Trip Report

Montreal, Canada

2026-07-14

Wonseok Oh
Korea University

1. FSE 2026

제가 처음 연구했던 PyTER를 22년도 FSE에 붙였었는데요. 재밌게도 제가 처음 맡은 후배의 첫 논문이 올해 FSE에 붙게 되었습니다. 22년도 FSE에서 좋은 기억들이 있었기에 이번 FSE도 기대를 갖고 참석 했습니다. 그러나 세상이 급변해버린건지 제가 바뀐건지 모르겠지만 올해 FSE는 썩 마음에 들진 않았습니다.

FSE는 재작년부터 발표는 선택사항으로 정책을 바꿨습니다. 듣자하니 저널화를 위한 정책이라고 합니다. 그런데 발표가 선택사항이 되다보니, 학회 전체의 퀄리티 컨트롤이 잘 안 된다는 느낌을 받았습니다. 어떤 세션은 발표자가 오지 않아 발표가 비어버리기도 했고, 어떤 발표는 발표자가 내용을 잘 모르니 다른 저자에게 이메일로 질문을 하라는 경우도 있었습니다.

사실 더 아쉬웠던 부분은 발표보다 아티팩트였습니다. 요즘 SE 연구는 기존 연구를 재현하거나, 기존 도구를 가져와 비교 실험을 하는 경우가 많습니다. 그런데 논문은 공개되어 있어도 실제 아티팩트가 공개되어 있지 않으면 다음 연구의 발판으로 삼기가 매우 어렵습니다. 저도 여러 발표를 들으면서 “이거 우리 연구에 써볼 수 있겠다”라는 생각을 하는데, 막상 아티팩트가 없거나 접근하기 어려우면 “이걸 써야하나”라는 부정적인 마음부터 듭니다.

특히 중국 쪽 연구에서 이런 경우를 몇 번 봤습니다. 아이디어 자체는 흥미롭고 결과도 좋아 보이는데, 도구나 데이터가 공개되어 있지 않아 실제로 써보기 어렵습니다. 이러면 논문은 분명 존재하지만, 커뮤니티 입장에서는 지식이 충분히 축적되지 못합니다. SE 연구가 점점 LLM, 대규모 데이터, 복잡한 실험 환경에 의존하게 되는 상황에서 아티팩트 공개는 선택사항이 아니라 연구의 일부로 봐야하지 않을까 싶었습니다.

22년도 FSE에 갔을 때는 “이 학회에 오면 이런 연구를 하는 사람들이 있구나”라는 설렘이 컸다면, 올해 FSE에서는 “이 연구를 내가 실제로 이어받아 쓸 수 있을까”라는 생각을 더 많이 했습니다. 아마 제가 연구를 바라보는 관점이 조금 바뀐 것도 있겠지만, 학회가 커지고 정책이 바뀌면서 생기는 부작용도 분명히 있는 것 같습니다.

1.1. 흥미로운 발표

그럼에도 불구하고 흥미로운 발표는 있었습니다. 기억에 남는 발표는 크게 세 가지였습니다.

1. Natural Language-Focused Software Engineering via Code-Documentation Equivalence [1]

주제는 주석과 코드가 서로 일치하는지 판단하는 연구였습니다. 간단하지만 굉장히 중요한 문제입니다.

흥미로웠던 점은 해결 방법이 굉장히 간단했다는 것입니다. 주석만 보고 LLM에게 코드를 생성하게 한 뒤, 생성된 코드와 실제 코드의 semantic이 같은지 비교하는 방식이었습니다. 즉 “이 주석이 정말 이 코드를 설명하고 있는가?”라는 문제를 “이 주석만 보고 만든 코드가 원래 코드와 같은 일을 하는가?”라는 문제로 바꾼 것입니다.

직관은 굉장히 simple한데 효과적이었습니다. 저는 이런 연구가 LLM을 잘 활용한 연구라고 느꼈습니다. LLM이 모든 것을 해결해준다는 식으로 접근하는 것이 아니라, 기존에는 애매하게 남아 있던 자연어와 코드 사이의 간극을 LLM이 잘하는 생성 능력으로 한 번 우회한 뒤, 그 결과를 다시 프로그램 분석 문제로 가져오는 방식이었습니다 (물론, semantic이 같은지도 LLM을 활용하긴 했습니다).

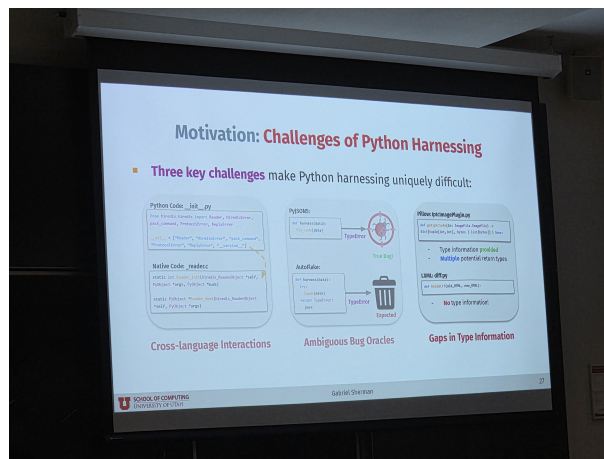
해결 방법이 단순한 대신, 이 프로그램이 얼마나 다양한 분야에 기여할 수 있는지 여러방면으로 결과를 보고했습니다. 이렇게 확실히 응용분야를 보여주는 점이 좋았던 연구였습니다.

2. SnakeCharmer: Automatic Fuzzing Harness Generation for Pure and Hybrid Python Libraries [2]

두 번째로 기억에 남는 발표는 Python 라이브러리를 위한 fuzzing harness를 자동으로 생성하는 SnakeCharmer였습니다.

이 발표가 인상 깊었던 이유는 제가 예전에 비슷한 문제에 부딪힌 적이 있었기 때문입니다. 저도 Python 테스트케이스를 자동으로 만들고 싶다는 생각을 했었습니다. 그런데 막상 해보면 문제가 한두개가 아닙니다. Python은 타입이 명시되어 있지 않은 경우가 많고, 어떤 입력이 유효한지 알기 어렵고, 테스트를 만들었다고 해도 그 결과가 맞는지 판단할 ground-truth를 얻기 어렵습니다.

그림 1 — SnakeCharmer가 지적인 Python에서 Fuzzing이 어려운 이유



저는 이런 문제들을 보다가 일단 포기했었습니다. 그런데 이 저자들은 그 지점에서 멈추지 않고, 문제를 끝까지 파고들어 좋은 프레임워크를 만들어냈습니다. 단순히 “LLM으로 테스트를 생성한다”가 아니라, Python 라이브러리에서 실제로 실행 가능한 harness를 만들기 위해 필요한 제약과 어려움을 하나씩 다룬 점이 좋았습니다.

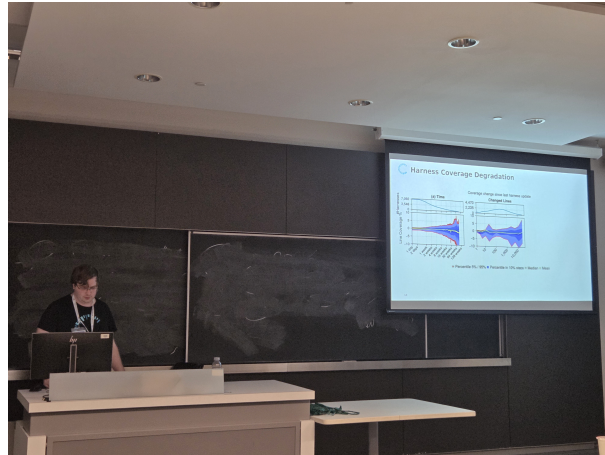
이 발표를 들으며 묘하게 반가운 기분이 들었습니다. 제가 혼자 고민하다가 막혔던 지점을 다른 연구자들도 비슷하게 고민하고 있었고, 누군가는 그 문제를 더 깊게 파고들어 논문으로 만들어 낸 것입니다. 연구를 하다보면 각자 자기 방에서 혼자 문제를 붙잡고 있는 것 같지만, 사실은 여러 연구자들과 알게 모르게 계속 소통하고 있다는 느낌이 들었습니다.

3. An Empirical Study of Fuzz Harness Degradation [3]

세 번째로 기억에 남는 발표는 fuzzing harness 도구들이 시간이 지나도 실제로 도움이 되는지 조사한 연구였습니다. 우리가 보통 도구 연구를 하면, 논문을 쓰는 시점의 벤치마크에서 얼마나 잘 동작하는지를 보여줍니다. 그런데 실제 소프트웨어는 계속 업데이트 됩니다. 오늘 잘 동작하던 harness가 몇 달 뒤에도 여전히 쓸모가 있을지는 별개의 문제입니다.

이 연구는 바로 그 지점을 practical하게 조사했습니다. 시간이 지나며 프로그램이 업데이트 되었을 때 기존 fuzzing harness가 충분히 잘 동작하는지, 그리고 여전히 coverage나 bug finding에 도움이 되는지를 살펴본 것입니다. 결론적으로는 fuzzing harness가 시간이 지나도 여전히 효과적이라는 결과를 보여주었습니다.

그림 2 — 시간에 따른 Coverage 변화 그래프



이 발표를 들으며 조금 신기했습니다. 우리가 만든 도구들이 논문이 accept되는 순간에만 반짝 쓰이는 것이 아니라, 프로그램이 바뀌고 시간이 흘러도 계속 도움이 될 수 있다는 뜻이기 때문입니다. 그런 점에서 이 연구는 도구의 지속가능성에 대해 생각하게 해주는 발표였습니다.

2. 메뉴

이번 FSE는 캐나다 몬트리올에서 열렸습니다. 캐나다에 대해 가장 크게 느낀 점은 자연이 좋다는 것이었습니다.

몬트리올에는 Mont Royal이라는 큰 공원(산일수도?)이 있습니다. 여기에는 큰 연못도 있어 산책하기 너무 좋았습니다. 연못의 이름은 Beaver Lake로 비버가 출몰한다는데, 저는 못 봐서 조금 아쉬웠습니다.

그림 3 — 자연이 좋았던 몬트리올.



자연이 좋아서 그런지 시장도 굉장히 잘 되어 있었습니다. 시장에 가보니 메이플 버터, 메이플 위스키, 메이플 식초, 메이플 커피 등 메이플 제품들이 정말 많았습니다. 한국에는 메이플스토리만 있는데 말이죠. 메이플이란 메이플은 싹쓸이 했는데 그렇게 사다보니 캐나다 국기에 단풍잎이 그려진 이유가 납득 되었습니다.

그림 4 — 메이플 제품이 가득한 시장.



근데 음식은 제 취향과 잘 맞지 않았습니다. 대표 음식으로 푸틴이라는 것이 있는데, 감자튀김에 이상한 소스를 올려둔 음식입니다. 캐나다까지 왔으니 한 번 먹어봤는데, 솔직히 맛이 없었습니다. 실제로 캐나다 사람들은 푸틴을 먹는지 궁금할 정도였습니다.

음식들이 입에 너무 맞지 않아 저는 금단의 결심을 하게 됩니다. 바로 한식을 먹는 것입니다. 몬트리올 도착한지 3일만에 순두부찌개를 먹으러 갔습니다. 솔직히 너무 행복했습니다. 한국에는 맛있는 음식들이 많다는건 정말 큰 축복임을 느꼈습니다.

그림 5 — 한식 뿐만 아니라 일식도 먹었다.



마지막으로 몬트리올은 재즈(Jazz)의 도시답게 재즈바가 많았습니다. 그래서 저도 한 번 재즈바에 놀러가봤습니다. 사실 이렇게 재즈를 본격적으로 듣는 것은 처음이었는데, 박자가 난해하고 곡의 흐름도 처음 맞닥뜨려서 그런지 노래가 조금 어색하게 들렸습니다. 아무래도 저는 재즈랑 잘 어울리지는 않나 봅니다.

그림 6 — 몬트리올의 jazz bar.



3. 마치며

이번 FSE 2026은 여러모로 생각할 거리를 많이 준 학회였습니다. 후배의 첫 FSE 논문이라는 기쁜 일도 있었고, 학회가 연구 커뮤니티로서 어떤 역할을 해야 하는지 다시 생각하게 된 계기도 생겼습니다. 또한, 연구로서 그치는게 아닌 실제 현실에 기여할 수 있는지도 얼마나 중요한지 생각해보게 되었습니다.

학회를 다녀올 때마다 느끼는 것들이 점점 많아진다는 걸 알게 되었습니다. 처음에는 발표들, 연구들을 보며 “이런걸 하는구나”였다면 지금은 “나는 어떻게 해야할까”라는 고민을 더 많이 하게 됩니다. 당장 답은 내릴 수 없지만, 이런 고민을 계속하다보면 언젠가는 답을 찾을 수 있지 않을까 생각합니다. 이렇게 좋은 경험을 할 수 있게 지원해주신 오학주 교수님께 감사의 인사 올리며 이상 Trip Report를 마치겠습니다.

참고문헌

- [1] A. Eghbali, Z. Liu, and M. Pradel, “Natural Language-Focused Software Engineering via Code-Documentation Equivalence,” *Proceedings of the ACM on Software Engineering*, vol. 3, no. FSE, pp. 4644–4665, 2026.
- [2] G. Sherman and S. Nagy, “SnakeCharmer: Automatic Fuzzing Harness Generation for Pure and Hybrid Python Libraries,” *Proceedings of the ACM on Software Engineering*, vol. 3, no. FSE, pp. 69–91, 2026.
- [3] P. Görz, J. Schilling, N. Bissantz, and T. Holz, “An empirical study of fuzz harness degradation,” *Proceedings of the ACM on Software Engineering*, vol. 3, no. FSE, pp. 3746–3768, 2026.