TRIP REPORT FSE 2022

22.11.13 ~ 22.11.18 SINGPORE

개요

FSE 2022 에 애정 어린 첫 논문 PyTER: Effective Program Repair for Python Type Errors 발표를 하기 위해 참여를 하게 되었다. 사실 FSE 2022 의 소감을 작성하기 전에 전달해야 할 메세지가 하나 있다. FSE 에 합격하게 된 계기 중 하나가 싱가포르에서 열린다는 점이었다. 오해하지 말아야 할 점은, 싱가포르가 아닌 다른 베뉴에서 개최 되었다면 지원을 하지 않았을 거라는 것은 아니다. '싱가포르에 가고 싶다'라는 작은 동기가 FSE 2022 에 도전할 수 있도록 이끌어 주었는데, 이것이 누군가에게는 보잘것없어 보이는 동기지만 나에게는 꽤나 큰 원동력이 되었다. 이 글을 읽는 모두에게 하찮고 작아 보이는 동기일지라도 그것이 자신을 이끄는 동력원이라면 그것은 소중한 마음가짐임을 알려주고 싶다.

이제 FSE 2022 에서의 발표는 어떠하였는지, 학회 분위기는 어땠는지, 그리고 배운 점은 무엇이었는지 작성하려 한다.

FSE 2022

나의 발표 세션

나에게는 한 가지 고민거리가 있었다. 바로 영어를 못한다는 점이다. 읽기는 어느 정도할 수 있지만, 읽기에 비해 듣기와 말하기가 한참 부족했다. 이를 극복하기 위해 학회 참석하기 3 개월 전부터 1:1 영어대화 학원을 등록했을 정도면 얼마나 큰 고민거리이고 급한 문제였는지 알 수 있을 것이다. 학원을 다닌 덕분에 말하기는 어느 정도 자신감이 붙었지만, 듣기는... 해결을 하지 못한 채로 싱가포르에 도착하게 되었다.



그림 1 발표는 막힘 없이 진행을 하였다

발표 자체에는 큰 두려움이 없었다. 발표가 15 분이었기에 스크립트를 열심히 만들고 외우기만 하면 문제가 될 일이 없었다. 그림 1 과 같이 발표는 부드럽게 진행 되었고, 나 자신도 발표가 끝나고 '잘 마무리 되었다'라는 느낌을 받을 정도였다.



그림 2 불안함과 초조함에 사로잡힌 모습이다

문제는 Q&A 세션이었다. 질문이 총 3 개가 들어 왔었는데 모두 처음에 듣기가 되질 않아 "I'm sorry?"로 시작하였다. 암쏘리로 시작하는 나 자신이 얼마나 부끄럽고 한심하였는지... 듣기를 못한다는 것을 감추기 위해 질문자의 말을 이해 못했다는 느낌을 주려고 "포인트를 이해하지 못하였다... 다시 한 번 설명해달라..."라는 식으로 대답을 하였는데, 이것이 큰 악수였다. 질문자가 핵심을 설명하기 위해 구구절절 풀어놓기 시작하는데 듣기가 되질 않으니 풀어 놓는 설명을 들었을리가 만무하다. 3~4 번의 암쏘리를 통해 질문 2 개는 어떻게든 대답을 하였지만, 나머지 질문 한 개에 대해서는 도저히 이해할 수 없어서 "세션이 끝나고 이야기 해보자"라고 대답을 해버렸다. 엉망진창이었지만 마무리 되었다는 사실에 안도감이 들었고, 다음 논문 발표 때는 이런일이 없었으면 좋겠다는 생각을 많이 했다.

그래도 발표를 진행하면서 느꼈던 점이 많았다.

첫번째로, 발표는 굉장히 즐겁다는 것이다. 연습을 하는 과정은 고되지만 고된 여정 끝에 광활한 대지를 볼 수 있듯이, 발표를 끝마치고 나면 뿌듯함과 자랑스러운 감정을 듬뿍 받을 수 있었다. 이 감정을 다시 느끼기 위해 멋있는 논문을 한 편 더 작성하고 싶은 마음이 생겨났다.

두번째로, 내 논문이 당당하면 학회의 사람들은 호의적이라는 것이다. 만약, 내 논문이 빈약하고 숨기려는 게 많은 상태에서 얼떨결에 학회에 붙었으면 Q&A 세션이 어떻게 보였을까? '듣기를 못해서 대답을 못하는구나'가 아니라 자꾸 감추려고 대답을 회피하는 모습으로 보일 것이다. 나는 내 논문이 당당했기에 학회 사람들은 호의적이었고 대답을 잘 하지 못했어도 모두 웃으면서 넘어갈 수 있었다. 힘들 때 웃는 자가 일류라는 말이 있다. 발표를 하면서 이것은 주객전도가 되었다는 것을 알 수 있었다. 일류만이 웃을 수 있는 상황이 나오는 것이다. 이것이 바로 당당하고 자랑스러운 논문을 작성해야 하는 이유였던 것이다.

마지막으로, 논문이 마무리 되었다는 것은 생각보다 더 큰 기쁨이라는 것이다. 처음 Accept 되었다고 글로만 보았을 때는, 물론 기뻤지만, 그렇게 오래가지는 않았다. 그런데 막상 학회장에 와서 발표를 하면서 내 논문이 붙었다는 사실을 몸소 느껴보니 훨씬 더 행복하고 내가 자랑스러웠다. 이 경험이 연구를 끝까지 진행하게 도와줄 것이고, 그 연구가 잘 마무리 되면 또 좋은 경험은 차곡차곡 쌓이게 될 것이다. 이러한 선순환에 빠져들기 위해 모두 힘들지만 끝까지 버티면서 논문을 작성하는 것이라고 생각이 들었다.

다른 분들의 발표 세션

다른 분들의 발표 세션에서 아쉬웠던 점은 나의 슬픈 듣기 실력이었다. 듣기가 제대로 되질 않아 제대로 이해한 발표가 손에 꼽았다. 그런데, 듣기가 처참해도 이해가 잘 되는 발표가 몇 가지 있었다. 대부분 문제 제기 혹은 동기에서 전달이 잘 되어 문제 해결 부분까지 부드럽게 발표가 진행된 케이스였다. 그렇기에 이번 FSE 2022 에서는 문제 탐색을 어떻게 해야하는지에 대해 많이 깨우친 학회였다. 도대체 어떤 식으로 문제 탐색을 하였기에 듣기가 처참한 나조차도 이해할 수 있었는지 몇 가지 논문을 통해소개하려고 한다.

LOCATING FAULTS WITH PROGRAM SLICING: AN EMPIRICAL ANALYSIS

이 발표는 '연구는 이렇게 접근 해야지'라는 느낌을 가장 크게 받았던 발표였다.

첫번째로, 문제 탐색이다. Fault Localization 에서는 두 가지 방법이 있다. 테스트 케이스를 이용하여 얼마나 Positive/Negative 케이스가 해당 line 을 통과했는지의 통계를 이용한 Statistical Debugging (SBFL 과 동일하다) 방법과, 에러가 발생한 지점(Assertion 에러일 경우, return 에서)부터 역으로 올라가 오류 line 을 찾는 Dynamic Slicing 방법이다. 문제 탐색을 위해 먼저 각각의 방법의 장단점을 파악하였다.

```
: covered statements
                                                  int middle(x, y, z) {
   ■: covered statements
                         x 3 1 3 5 5 2
                                2 \ 5 \ 3
   int middle(x, y, z) {
                         y 3 2
                                               2
                                                      int x, y, z;
                                                                           z 3
2
       int x, y, z;
                         z 5 3 1
                                  5 4 3
                                               3
                                                      int m = z;
                                                      if (y < z) {
       int m = z;
                           4
       if (y < z) {
                           -----
4
                                          4
                                                         if (x < y)
                                               5
          if (x < y)
                           5
                                                             m = y;
             m = y;
                                                         else if (x < z)
                                               7
7
          else if (x < z)
                                                            (m)
                                               8
             m = y;
                                               9
       } else {
9
                                               10
                                                             (x > y)
                                                                               10
          if (x > y)
                                               11
                                                             m = y;
                                                                              11
                                         11
11
             m = y;
                                                         else if (x > z)
12
          else if (x > z)
                                          12
                                               12
                                                            m = x;
                                                                             □ 13
13
             m = x;
                                         13
14
                                       14
                                             14
                                                                              14
15
       return m;
                                    ■ 15
                                             15
                                                      return(m;
                                                                             15
   }
16
                                              16
```

그림 3 Fault Localization 방법 두 가지

Statistical Debugging 에는 두 가지 장점이 있다. 오류 지점을 찾았다면 Top ranked 일확률이 높다는 것과 Multiple Faults 를 잘 찾아낸다는 것이다. 다만, 테스트 케이스의 의존하기 때문에 테스트 케이스가 적절하지 않으면 효율은 굉장히 떨어진다.

반면에, Dynamic Slicing 은 Single fault 를 찾는데 매우 효과적임을 알아내었다. 게다가 오류 케이스만 있어도 효과적으로 오류 지점을 찾아낼 수 있다. 조사 결과를 보면 두 가지 방법의 장단점이 서로 반대임을 볼 수 있다. 그렇다면, 두 가지의 장점만 골라 적절히 합친다면? 이것이 이 논문의 문제 탐색이자 문제 정의의 시작이었다. 막연하게 상상을 펼친 것이 아니라 조사를 통한 확실한 근거를 도출했기에, 적절하게 합치는 방법만 알아내면 연구의 결과는 보장 되어있는 것이다. '연구는 이렇게 시작해야지'를 느낄 수 있었던 논문이었다.

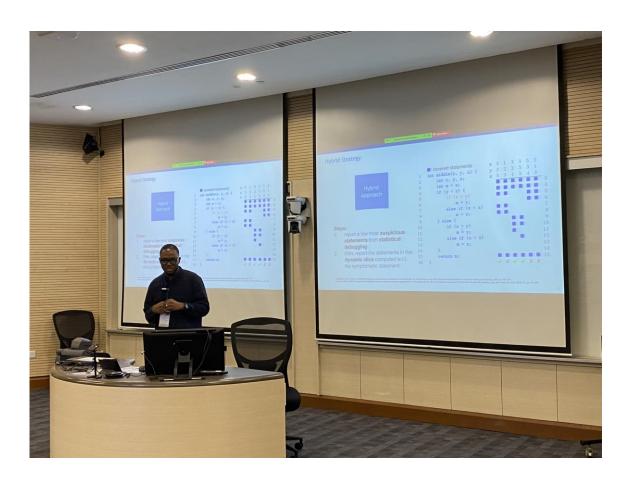
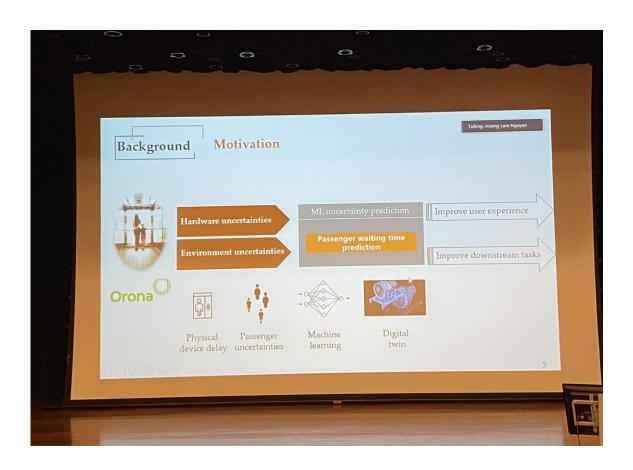


그림 4 문제 제기가 깔끔한 발표였다

다만, 아쉬웠던 점은 Empirical Analysis 여서 적절하게 합치는 방법은 Baseline 에 불과하다는 것이었다. 아마 조사를 끝냈으니 후속 연구로 진행하고 있을 것이 분명하다.

UNCERTAINTY-AWARE TRANSFER LEARNING TO EVOLVE DIGITAL TWINS FOR INDUSTRIAL ELEVATORS

이번에는 실생활 문제(엘리베이터 학습)를 가져와 해결한 논문이었다. 전세계 사람들 중 엘레베이터를 타지 않은 사람은 아마 없을 것이다. 그렇기에 문제가 굉장히 명확했고 충분히 공감을 살 수 있는 문제였다.



문제는 다음과 같다. 엘레베이터가 적절하게 사람들을 태우고 내리기 위해, 현재 상황에서 어떤 선택을 해야할지 판단을 내려야 한다. 적절한 판단을 내리고 있는지 확인하기 위해, 현실 세계를 컴퓨터로 디지털화 시켜 시뮬레이션을 통해 결과를 예측하는 Digital Twin 이라는 기술을 사용하게 된다. 그러나, 엘레베이터의 성능과 작동 방식은 나날이 바뀌어 가고 있다. 따라서, Digital Twin 또한 엘레베이터의 환경에 맞게 적절하게 변형이 되어야 할 것이다. 문제는 여기서 나오게 된다. 엘레베이터의 환경이 바뀔 때마다 사람이 수동으로 Digital Twin 환경을 바꾸어 왔는데, 이것을 ML을 통해 자동화(적응) 시키겠다는 것이다.

문제는 명확했기에 해결해야 할 점도 명확했다. 어떻게 환경이 바뀌었는지를 알아내어 그것에 맞추어 시뮬레이션을 만들면 된다.

환경이 매번 바뀌는 것을 Uncertainty 라고 정의한 후, 이것을 수치화 시키는 아이디어인 Uncertainty Quantification 이라고 명명하였다. Pre-Trained 모델이 있으면 엘레베이터 환경에 맞추어 Transfer Learning 이 진행된다. Transfer Learning 에서 사용한 데이터 중, Pre-Trained 모델이 제대로 활용하지 못하는 데이터(이것이 바로 Unceratainty)를 판별하여 이 데이터에 집중하여 학습을 하겠다는 것이다. 모두가 공감하는 실생활 문제였고 무엇을 해결해야 할지가 명확했기에 아이디어 자체는 간단해 보여도 임팩트 있는 발표로 다가왔다.

여담으로, 내가 발표한 PyTER 도 이러한 케이스에 속하는 것 같다. Python 에서의 Type Error 는 모두 겪는 문제였기에 무엇을 해결해야 할지 명확했다. 그렇기에 간단한 아이디어일지라도 해결만 할 수 있다면 바로 논문으로 완성될 수 있었다. 그렇게 완성된 것이 이번 FSE 2022 에 제출한 PyTER 인 것이다.

ADAPTIVE FAIRNESS IMPROVEMENT BASED ON CAUSALITY ANALYSIS

이 발표는 연구 내용의 전달보다는 PPT를 잘 만들어야 하는 이유를 느꼈던 발표였다. "논문 내용만 훌륭하면 되지, 발표 자료까지 훌륭해야 돼?"라고 생각하는 분들이 있다면, 반드시 그 생각을 꺾어야 한다고 생각한다. 발표 자료가 깔끔하고 명료하다면 저자가 자신의 연구를 모두 이해한 상태로 정리를 했단 느낌을 줄 수 있다. 그런데, 발표 자료가 너저분하고 핵심이 없다면 괜히 '연구를 똑바로 한 게 맞나?'라는 의구심이 들정도로 부정적인 감정을 주게 된다.

이 연구는 러닝을 할 때 데이터의 Fairness 를 해결하는 연구로써 얼마나 Fairness 가 개선 되었는지를 한 눈에 알아볼 수 있도록 발표가 진행 되었어야 했는데...

그림 5 처럼 딱 봐도 슬라이드가 한 눈에 들어오지 않는다. 특히, 축의 설명도 없고 그래프만 잔뜩 있어서 이 논문의 핵심을 파악하기가 어려웠다.

도표 자체는 논문의 도표를 가져왔고 논문에서도 도표의 축 설명은 없었다. 당연히 논문에서는 글로 축에 대해서 설명을 해놓았을테지만, 발표에서는 청자가 발표 시간 내내 집중할 수 없기에 잠깐 딴 생각을 하다 와도 이해가 될 수 있을 만큼 발표 자료가 완성 되어있어야 한다고 생각한다. 그림 5 와 같이 만든다면... 발표에 대한 흥미도 떨어지고 대충 준비해왔다는 부정적인 생각이 많이 들 수 밖에 없다. 이 글을 읽는 모든 분들은 이러한 불상사가 없길 바라며 해당 항목을 추가하였다.

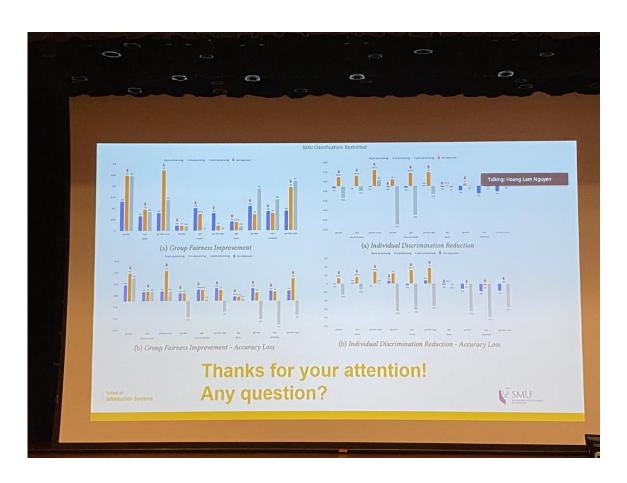


그림 5 축의 설명도 없고 그래프를 4 개나 넣어놔서 알아보기 힘든 상태이다

싱가포르

개요에서 설명했 듯이 FSE 2022 에 무사히 제출할 수 있었던 계기 중 하나가 싱가포르에 오고 싶어서였다. 서양인들의 휴양지라고 불릴 정도로 해외에서 인기가 많은 나라라고 하길래 얼마나 휴양하기 좋으면 이러한 말이 있었는지 궁금하였다.

다녀와 본 결과, 싱가포르는 휴양지가 가져야 할 덕목을 모두 갖추었음을 알 수 있었다. 여기서 중요한 점은 여행지가 아니라 휴양지라는 점이다. 내가 정의하는 휴양지는 말그대로 '쉬러 오는 곳'이라고 생각하면 될 것 같다. 이러한 정의 아래서, 내가 생각하는 휴양지가 가져야 할 덕목은 바로 '즐길거리가 얼마나 많은가'이다.

즐길거리들

싱가포르는 즐길거리가 너무 많았다. 풍경 뿐만 아니라 체험할 수 있는 것도 굉장히

많았다.



매일 때가 되면 레이저 분수 쇼가 진행되었고, 싱가포르의 랜드마크 중 하나인
가든즈베이에서도 노래에 맞추어 건물들이 번쩍이는 쇼가 진행 되었다. 모든 여행객이 이 쇼를 보러 오는데, 특정 날이 아니라 매일 진행되기 때문에 인파가 몰리지 않아서 쾌적하게 볼 수 있었다.

다만, 아쉬웠던 점이 두 가지 있었다. 레이저 분수 쇼는 크루즈를 타면서 볼 수 있다고 하였는데 배의 운항 시간이 코로나로 인해 바뀌어 체험하지 못하였던 것과, 가든즈베이 쇼는 누워서도 볼 수 있다는데 장소를 잘못 골라 그러지 못하였던 점이었다. 그럼에도 불구하고, 싱가포르에서만 볼 수 있는 진귀한 광경을 볼 수 있어서 싱가포르에 왔음을 몸소 느낄 수 있었다.

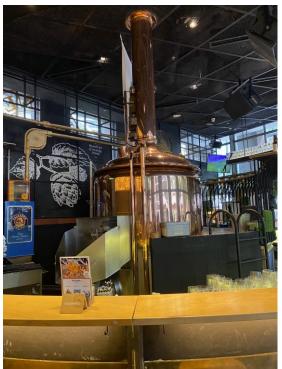
그 뿐만이 아니라, 싱가포르에는 나이트 사파리로 유명한 동물원도 있었다. FSE

2022 에서 주도하여 학회 사람들과 다 같이 가게 되었다. 사람이 너무 많다 보니 진행이 원활하지 않아 따로 와보고 싶은 마음이 들긴 했었지만, 여러 동물들을 볼 수 있어서 진귀한 경험이었다.



싱가포르 동물원에는 흔히 볼 수 없는 동물들이 많으니 꼭 낮에 들려서 밤까지 즐기는 것을 추천한다.





싱가포르는 강을 하나 끼고 있는데, 그 강 사이를 잇는 DNA 교가 있다. 여기서 가든즈베이가 보이는 뷰는 일품이어서 사진도 한 장 건졌다. 휴양지 느낌이 물씬나게 가져온 하와이안 셔츠 한 벌이 빛을 발하는 순간이었다.

또한, 뜬금없이 파울라너 양조장도 있길래 가보았는데 거대한 배럴이 있어서 놀랐다. 구글의 평도 좋았고 딱 봐도 맛있어 보이길래 얼른 들어가 맥주를 한 잔 하였다. 싱가포르는 기온이 매우 높은 곳이어서 밖을 돌아다니면 땀을 뻘뻘 흘리게 되는데, 이 상태에서 맥주 한 잔을 들이키면 지상낙원을 대가 없이 즐길 수 있었다. 심지어 Happy Hour 까지 겹쳐서 어쩔 수 없이 맥주를 두 잔 마셨었다.

마치며

이번 학회 참석은 여러모로 뜻 깊은 시간이었다.

- 논문 발표를 처음으로 한 것
- 연구의 문제는 이렇게 잡는 것이다!
- 싱가포르를 어떻게 즐길 것인가?

이러한 값진 경험을 할 수 있도록 많이 도와주신 오학주 교수님께 감사드리고, 다 같이 가서 즐거운 시간을 보낼 수 있도록 해준 연구실 분들께도 감사의 인사 드립니다.