SeungHyun Park(박승현)

Birth : xxxx.xx.xx. ¦ Tel : (+82) 010-xxxx-xxxx ¦ E-mail : xxxxxxx@naver.com
Blog : https://velog.io/@koohack ¦ GitHub : https://github.com/koohack

EDUCATION

성균관대학교 공과대학 & 소프트웨어대학 March 2016 - February 2022
공과대학 기계공학부 학사 졸업
소프트웨어대학 컴퓨터공학과 학사 졸업 (복수전공)
Cumulative GPA: 3.64 / 4.50
Computer Engineering GPA: 4.13 / 4.50
Mechanical Engineering GPA: 3.64 / 4.50

Publications

1. **Finding Cryptographic Misuse in Chinese Android App,** Seunghyun Park, CISC W, 2021 December 2021

WORK EXPERIENCE

LG innotek - Vehicle SW Development (R&D) July 2021 - August 2021
Intern

- **NAND에 발생한 오류를 자동으로 탐지하는 도구를 python Qt로 rule base로 구현**
  전문가가 아니여도 빠르게 해당 도구를 사용할 수 있도록 친숙한 UI를 제작했다. 이 과정에서 현직자와 실제 사용할 유저의 피드백을 받고 UI도 개선해보고 새로운 기능도 추가하여 유저 편의성을 증진시켰다.

- **오류 탐지율 95%로 향상 및 오류 검출을 시간 1 h에서 1 min으로 단축**
  기존에 오류가 발생한 경우, 전문가가 오류가 발생한 부분을 정적으로 분석하여 오류를 찾아내는 반면, 이 도구를 활용함으로써 빠르게 오류를 탐지할 수 있게 됐다. 또한, 분석 속도를 개선하기 위해 bad block 탐색 알고리즘을 새로 고안하여 기존 알고리즘보다 10% 정도의 성능 향상을 이뤄냈다.

- **ML/DL 방법론을 적용해 오류를 사전에 탐지 가능한 방법론 연구**

SungKyunKwan University Seclab - Researcher March 2020 - March 2022
Research Intern

1. AFL Fuzzer를 사용해 ChibiOS 내 버그 탐지

- **Fuzzer 관련 논문 학습 및 내용 정리**
  The Art, Science, and Engineering of Fuzzing: A Survey 논문을 읽고 다양한 fuzzer에 대한 이해도 향상시키고 ChibiOS에 적용 가능할만한 fuzzer를 탐색했다. Fuzzer는 대부분 open source 형태로 제공됐기에 이를 동작시키는 과정에서 시스템의 동작 과정을 학습할 수 있었다.

- **ChibiOS가 가벼운 OS이기에 AFL seed, mutation 등을 변경했지만 버그 탐지 실패**
  ChibiOS는 다른 OS와는 달리 light weight OS이기 때문에 복잡한 함수가 거의 존재하지 않는다. 따라서 seed와 mutation 부분을 변경하더라도 dynamic한 변화를 이끌어 내기가 어려웠다. 또한, AFL을 제외한 fuzzer를 사용해보려 했으나 ChibiOS 특성상 다양한 input을 받지 않을뿐더러 구조도 단순했기에 다른 fuzzer를 적용하기 보다 보편적으로 사용되는 AFL만으로 충분하다고 판단했다. fuzzer의 특성상 1차례의 실험을 진행하는 과정에서 많은 시간이 소요됐지만 ChibiOS의 light weight한 특성상 fuzzer로 bug를 찾아내기는 쉽지 않았다.

- **Isabelle/HOL을 적용해 ChibiOS에 논리적 오류 발생 여부 연구**
  Isabelle/HOL이라는 논리 언어를 통해 C++로 구현이 된 ChibiOS 코드를 변경하고 코드 자체에 논리적 결함이 있는지 판단하는 실험을 진행했다. Isabelle/HOL 언어에 익숙해지기 위해 연구실에서 진행하는 개별 스터디에 참석했었다. Secure OS 프로젝트에 참여했었는데, 이 과정에서 ChibiOS code의 10% 정도 분량의 코드를 Isabelle/HOL 언어를 통해 번역 후 분석을 진행해봤다.

2. 안드로이드 앱 암호화 오용 연구

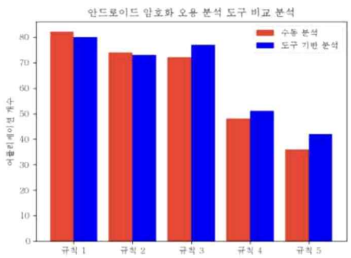
(Github : https://github.com/koohack/Crawler_CN_App_Store)

- **중국 app store로부터 안드로이드 앱 크롤링 가능한 툴 개발**

  중국에서 가장 많이 사용되는 앱 스토어 2곳에서 앱을 BeautifulSoup4(Python)을 사용해 크롤링했다. 광범위한 분석을 진행하기 위해 10개의 카테고리에서 랜덤으로 25개 어플리케이션을 추출해 분석을 진행했다.

- **Reverse Engineering을 통해 암호화 오용 어플리케이션 분석 (250개 어플리케이션 중 196개(78.4%) 암호화 오용)**

  총 250개 어플리케이션에 대해 5가지 암호화 오용 기본 규칙을 위반했는지 여부를 분석했다. 분석 과정에서는 JADX tool을 사용했으며, reverse engineering된 java 코드를 분석하면서 암호화 오용 여부를 판단했다. 각 규칙에 접촉된 어플리케이션 수는 136개, 109개, 118개, 80개, 69개이다. 자동 분석 도구와 수동 분석 방법을 비교해 본 결과, 수동과 자동 방식의 오류를 고려했을 때 큰 차이는 발생하지 않은 것을 확인할 수 있었다.



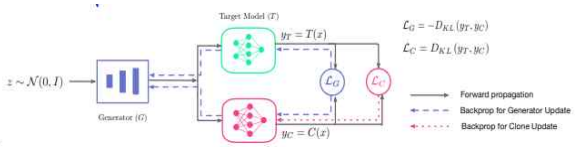
- **중국 어플리케이션의 개인정보 유출 정황 포착**

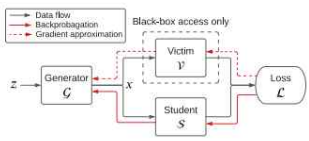
  모든 어플리케이션의 코드를 들여다보는 과정에서 많은 어플리케이션이 특정 기업에게 유저의 정보를 공유하는 API를 사용하고 있는 정황을 포착했다.

3. Model Stealing 연구

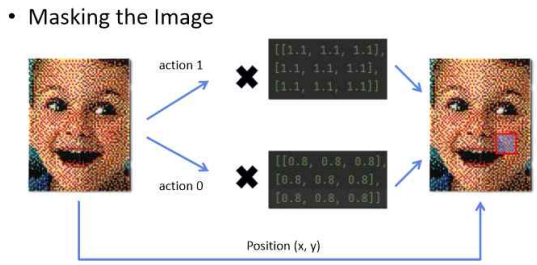
- **Block box 형태로 가정되는 CIFAR-10으로 학습된 DL 모델을 유사한 성능을 내는 모델로 Clone하는 실험 진행**

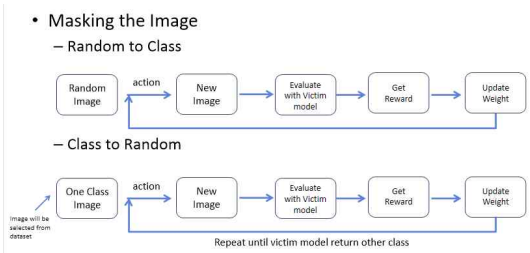
  기존 대부분 연구는 GAN 형식으로 generator를 학습시켜 replacable train data를 생성했다. 이 방식의 문제점은 GAN의 단점들을 많이 포함하고 있다는 점이었는데, 특히 train을 잘못시키는 경우 특정 class에 대한 편향이 커질 수 있었다. 또한, 이 방법론들은 Black box 형태로 query하여 모델을 clone하는 것인데 query의 수도 2M~40M으로 매우 컸고, PRADA라는 defence 방법론에 의해 탐지되기 매우 쉬웠다.





  GAN의 단점을 보완하고 PRADA 방법론을 우회하고자 강화학습 기반 Model stealing 기법을 연구했다. 제시한 방법론으로는 Image Masking 방법이었다. 현재 이미지를 encoder에 입력하고 action을 취할 position을 받는 형태로 진행하는 것이다. 또한, 이 방법론은 2가지 방식으로 분리할 수 있는데 한 방식은 random image에서부터 image를 변경해 나가는 것이며, 두 번째 방식은 완벽한 image에서부터 시작하는 것이다.





- **기존 GAN based 방식이 아닌 강화학습 방법으로 Latent vector를 표현해 train input 생성 (CIFAR-10 중 4개 Class에 대한 input은 적절히 생성됐으나 다른 Class에 대한 성능 측정 불가)**

  이 방법론 또한 특정 class에 특정되는 image를 생성해내는 문제점이 발생했다. 이는 class에 대한 정보가 적기 때문에 발생하는 문제라고 파악했다. 따라서, 변경된 image의 loss에 penalty 혹은 advantage를 주기 위해 중간에 classification model을 추가해 실험을 진행했다.

4. Unity 로봇팔 강화학습 적용

- **Unity로 3자유도 로봇팔을 제작하고 강화학습을 적용해 원하는 위치에 공을 던지는 실험 진행**
  Unity를 사용해 간단한 로봇팔을 제작했다. 복잡한 실험이 아닌 적용 가능성을 보기 위한 실험이었기에 3자유도 로봇팔로 제작했다.

- **DQN 방식으로 학습을 시켰고, 원과의 거리가 1 블록(Unity 내 약 50 size) 내인 경우가 60% 2 블록 내인 경우가 82%인 강화학습 모델 구현**
  로봇팔의 각도, 각속도 등 여러 요소를 input으로 넣고 가속할지 감속할지 여부를 결정한다. 로봇팔이 빨간색 점에 공을 정확하게 던질 수 있도록 실험을 진행했다. 그 과정에서 DQN 뿐만 아니라 사람의 행동을 모방하는 DQfD 방식도 적용해봤다.

Extra Learning

Boostcamp AI Tech 4th – NLP Tack　　September 2022 – Present

(관련 documents github에 저장)

1. STS, RE, MRC 등 NLP 모델링 대회

(Github : https://github.com/boostcampaitech4lv23nlp1/level2_klue_nlp-level2-nlp-06)
(Github : https://github.com/boostcampaitech4lv23nlp1/level2_mrc_nlp-level2-nlp-06)

- **기존 text classification approach에서 활용하는 [CLS] token이 아닌 multi-sentence + [MASK]를 활용해 pre-train 형식과 유사한 train 방식 고안** ([CLS] 토큰 대비 1-2% 성능 향상)

- **KLUE RE 데이터에 문제점이 있다는 것을 파악하여 적절한 기준을 정한 이후 전체 데이터에 대한 전수검사 진행 및 데이터 relabel & retype** (3만여 데이터 중 총 1170 건의 오류 발견, 최종 성능 소폭 상승)

2. 데이터 제작

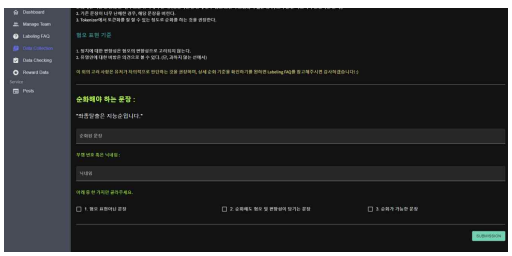
(Github : https://github.com/boostcampaitech4lv23nlp1/level2_dataannotation_nlp-level2-nlp-06)

- "예능" 관련 데이터 수집 및 Labeling, "예능" 데이터 Entity 관계 정의, 총 3회 Pilot tagging, 2회 labeling, 2회 검수를 통해 1664개 문장-label 쌍 추출 (Fleiss' Kappa 0.945 달성; bert based 학습 결과 micro f1 88.095%)

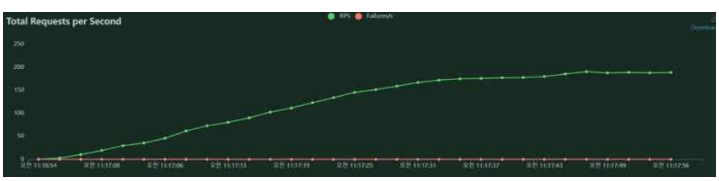
3. 혐오 및 욕설 댓글 순화 프로젝트

(Github : https://github.com/boostcampaitech4lv23nlp1/final-project-level3-nlp-06)

- 욕설에 대응되는 순화 데이터를 편하게 수집하기 위한 웹 페에지 제작(React, FastAPI)



- ChatGPT에서 추구하는 사람의 의도대로 학습되는 모델 구현을 위해 순화 모델에 "InstructGPT" 적용
- 한정된 리소스 내에서 서비스하기 위해 locust 사용해 backend stress test 실시 및 backend 구조 최적화



LGAimers – ML/DL **July 2022 – August 2022**

- Dacon "자율주행 센서의 안테나 성능 예측 AI 경진대회" 참여, Sklearn 라이브러리 내 모델과 custom deep learning 모델을 활용해 Regression model 구현 (XGBoost의 hyperparameter를 tuning하고 데이터 노이즈 제거 및 전처리를 통해 5% 성능 향상 달성) **(TOP 5%)**

Projects

Dacon 문장 유형 분류 AI 경진대회- NLP **December 2022 – December 2022**

- 제공된 데이터의 Imbalance를 해결하기 위해 Focal loss, over-samling 등 여러 기법을 사용해 실험 진행
- 다양한 방식으로 model을 fine-tuning 실시, Prompt Learning 기반으로 학습된 모델의 특성에 맞게 학습 방식 변환
- 총 335팀 중 11등 (1등과의 F1 score 차이 0.005(0.5%))

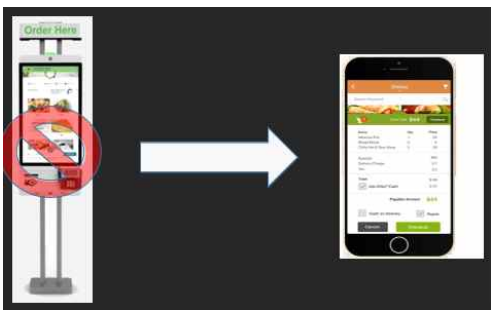
OIDC 클라우드 개발 경진대회 – Software Development **May 2022 – July 2022**

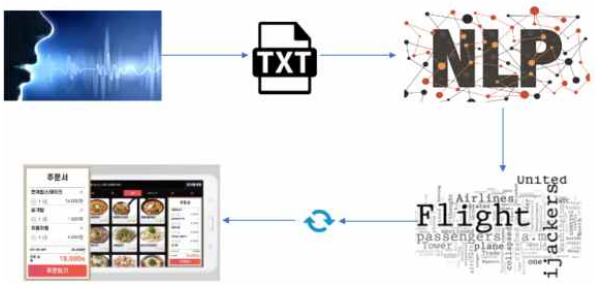
- 문과 및 예대를 위한 협업 겸 기여도 평가 플랫폼 개발
- FastAPI로 Backend, React로 frontend를 개발 환경은 네이버 클라우드 플랫폼 사용

NLP based Smart Kiosk – Software Development, NLP **September 2021 – December 2021**

(Github : https://github.com/koohack/Menu_Extraction_NLP)

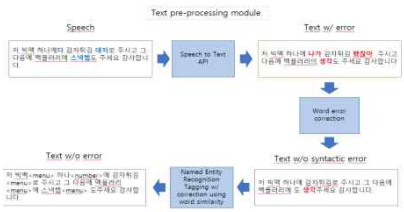
- 음성만으로 주문하는 플랫폼 개발

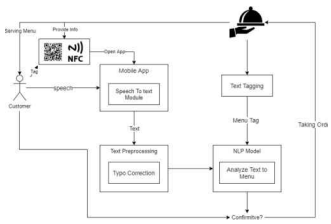




- 주문 관련 데이터 수집 및 제작 (AIhub 데이터 + 개별 데이터 제작)

- 오픈소스 STT의 오탈자를 수정하기 위한 알고리즘 구현 (실험 결과 오탈자 수정률 92%)





- 배달 기업에서 API를 제공해주지 않기에 유일하게 웹 서비스를 제공하는 요기요 웹 서비스와 크롤링(Selenium) 기반으로 연동할 수 있는 시스템 구현