AX 실무 기획론

건국대학교 기술경영학과 최보규

기술과 실무 사이에는 분명한 간극 존재

```
- AI기술이 워낙 복잡하고, 빠르게 변화하기 때문
- 대부분의 기획자들이 어디서부터 시작할지, 무엇을 먼저 도입할지 갈피를 잡을 수 없음
```

- 책을 통해 "실제 서비스에서 어떻게 사용되는지 정리" 후 간극 줄이기

- AX는 기술보다 설계가 훨씬 중요

- How to make AI가 아니라 What to do with AI, How to use AI에 초점

1. 회원가입

정의

고객이 서비스와 처음으로 맞닥뜨리는 순간

- 이 단계에서의 경험이 첫인상 효과와 동일하게 작용 -> 이후 상호작용에 긍정적 영향
- 고객 분석, 추천, 개인화 전략에 필요한 **데이터 확보**

AI 기술 **자연어 ID추천**, **데이터기반 ID추천**, **생성형 AI기반**

자연어처리(NLP[1])ID추천

- 말 그대로 자연어 처리 기술로 고객이 입력한 값을 분석해서 핵심 키워드를 추출
- 추출한 정보로 조합, 변형하여 아이디 대안을 제시
- "토큰화" :: 키워드를 의미 단위로 분리하는 것
- 사용자 입력 ▶ 성과 이름 분리, 이니셜 생성 ▶ WorldNet[2]활용 무작위 단어 생성 ▶ FastText[3]로 키워드 조합 ▶ Elasticsearch, PostgreSQL활용 아이디 중복 검사
  ex) Bogyu Choi ▶ Choi, Bo, Gyu & C.B.G ▶ Greeb, eco, time, pro 생성 ▶ C_pro, Bogyu_Green ▶ 중복검사

데이터 기반 ID추천

- 사용자의 입력 데이터는 물론, 다른 사용자들의 ID패턴, 요즘 검색 트랜드, 인기 키워드 등을 분석 회 최적의 ID 제공
- 사용자 입력 ▶ 성과 이름 분리, 이니셜 생성 ▶ Elasticsearch, PostgreSQL활용 기존ID와 유사한 ID 찾고 Scikit-learn사용해서 패턴 추출 ▶ Google Trends API, Twitter API로 키워드 트랜드 분석 ▶ FastText, Word2Vec 이용 유사ID 패턴 추천 ▶ 아이디 조합 ▶ 사용자 선택

ex) Bogyu Choi ▶ Choi, Bo, Gyu & C.B.G ▶ Bogyu_gaming, Gamer_Choi▶ Game 분석 ▶ Game=esport, winning, pro, gamer ▶ Pro_Bogyu, Bogyu_Gamer ▶ 사용자가 선택

생성형 AI기반 ID추천

- GPT기반 생성형 AI가 창의적인 ID를 자동 생성.
- 기존에 없던 독창적인 조합을 생성
- 사용자의 개성을 반영할 수 있음
- Hugging Face Transformers 라이브러리의 사전 학습된 AI 언어 모델들 사용 ▶ 토큰화 ▶ BERT, RoBERTa모델 활용 문맥 파악 ▶ ID후보 생성 ▶ 추가 정제과정
- ex) Bogyu Choi ▶ Choi, Bo, Gyu & C.B.G ▶ AI가 판단후 kor, *zi존*, bbo_이런 단어 생성▶ Bbo_Bogyu 이런거 하면 발음하기 유용하고 문맥적으로 기억하기도 쉬움 ▶ Bbo_bogyu, Zi존_bogyu, Kor_Choi 생성 ▶ 사용자와 인터랙션 후 최종 결정

AI 기술 비밀번호 추천 "AI기반 난수 생성", "암호화 보안 유지"

기존 비밀번호 생성규칙

- 최소길이
- 대소문자 포함
- 숫자, 특수문자 포함
- 연속문자 사용제한

AI기반 난수생성 -- 보안성 vs 편의성 --

- Python secretes[4] 라이브러리 사용
- LSTM[5], GAN[6] 모델 사용
- 취약 패턴 학습 및 방지 - 머신러닝 사용
- 최적 비밀번호 생성

암호화 저장

- Python bcrypt[7] 라이브러리 해싱[8]
- 솔트 추가
- 레인보우 테이블[9] 방어
- 로그인시 사용자가 비밀번호를 입력하면 저장된 솔트를 찾아 입력된 비밀번호에 추가하고 해시 함수에 넣어서 해시값을 비교해보는 식으로 작동
- 원본 비밀번호를 직접 저장하지 않고 검증 가능

신용카드 등록 "이미지 전처리", "텍스트 추출"

이미지 전처리

- 입력받은 하나의 이미지를 크기가 동일한 여러개의 정사각형 이미지로 쪼갠다 ▶ 이미지를 구성하는 색을 입력하는 방식을 통일한다.

- Open CV [10] 사용 이미지 대비, 기울기 등 보정 ▶ 이미지 품질 높이기.
- Tensor Flow EAST[11] 모델 활용 카드 영역 자동 감지

텍스트 추출(OCR[12])

- 촬영 장비(휴대전화 단말기)에서 OCR처리를 거친다.
- 사진 속 카드번호, 만료일, 카드사 로고와 같은 정보들이 텍스트로 인식된다.
- 주의사항 : 일반 문서정보와 달리 신용카드 정보는 인터넷을 통해 클라우드 서버로 전송시키게 되면 보안상의 문제가 발생할 가능성이 높다. 따라서 오픈소스 API[13] 사용은 주의해야 한다.
- 따라서 정보 보호를 위해 외부 전송이 되지 않는 장치 내에서 처리되는 OCR 기술을 사용해야 한다.

유효성 검증

- Luhn 알고리즘[14]을 사용하여 인식된 정보들이 유효한지 검증한다.
    1. 카드번호 오른쪽에서 왼쪽으로 숫자를 가져온다.
    2. 오른쪽부터 짝수번째 자리의 숫자를 두배로 만든다.
    3. 두배한 숫자가 10 이상이면 각 자릿수를 더한다.
    4. 모든 숫자를 더한다.
    5. 합계가 10의 배수가 되기 위한 검사 숫자를 계산한다.

EX) 카드번호가 1789372997 라면

1단계	**7**	**9**	**9**	**2**	**7**	**3**	**9**	**8**	**7**	**1**
2단계	14	9	18	2	14	3	18	8	14	1
3단계	5	9	9	2	5	3	9	8	5	1
4단계	모든	숫자의	합	:	56					

5단계 : 56과 가장 가까운 10의 배수는 60 따라서 4가 검사 숫자

최종결과 : 신용카드 번호 = 1789372994

신용카드 정보 조회

- 유효성 검증을 마친 신용카드 번호를 카드 데이터베이스 조회를 통해 어느 은행의 어느 카드인지 확인한다.
- Scikit-learn[15]을 사용하여 부정사용, 도난카드 등록, 비정상적인 패턴 감지

- 필요한 데이터
    1. 신용카드 정보 : 카드 발급 국가, 카드 유형, 신용카드 번호 등
    2. 사용자 정보 : IP주소, 사용자 단말기 정보, 웹 브라우저 정보 등
    3. 위치 데이터 : 사용자가 카드를 등록하거나 사용한 지역
    4. 행동 데이터 : 동일IP, 단말기, 웹 브라우저에서의 등록 시도 등
- 데이터들을 통해 사용자가 일정 기간 내에 몇 번이나 카드를 등록했는지, 사용자가 과거에 등록했던 위치와 현재 위치가 일치하는지, 과거에 등록하는데 걸린 시간과 크게 차이가 나는지 등의 패턴을 분석해 이상치를 탐지한다.
- 이상치를 탐지하면 Face ID, 지문인식, 생체인증, OTP인증을 요구.

신분증 인증 "객체 탐지", "위변조 검증"

객체 탐지

- OpenCV로 신분증, 얼굴 등 이미지 전처리 수행
- YOLO[16]활용 신분증 객체 탐지
- Tesseract OCR[17] 이용 신분증 텍스트 인식 후 추출

위변조 검증

- CNN[18] 보안요소 패턴 분석- 홀로그램, 보안 패턴, 미세한 특수 무늬 잘 있는지 확인
- 딥페이크 탐지 : 사진 합성 여부 확인 - 얼굴 확인
  ExifTool : 이미지 메타데이터 분석 -
- 신뢰도 점수 추출(0~1)
  - 0.95 = 신분증이 진짜일 가능성 매우 높음
  - 0.40 = 신분증이 위조일 가능성이 높음
  - 0.10 = 신분증이 위조됨:
  위변조 검증에 가장 중요한 핵심 기술은 딥러닝 모델 💡 딥러닝이란?

---

1. Natural Language Processing ↩
2. 프린스턴 대학교가 개발한 의미를 지닌 단어의 네트워크. 유의어 집합을 통해 단어 간 관계를 구조화한 데이터베이스. ↩
3. Facebook AI Research에서 개발한 단어 표현, 텍스트 분류 라이브러리. 대규모 텍스트 처리에 효율적 ↩
4. 암호학적으로 강력한 난수 생성 가능 - 비밀번호, 토큰 생성에 유리 - ↩
5. Long Short-Term Memory ↩
6. Generative Adversarial Network ↩
7. 솔트를 자동으로 적용하고 의도적으로 연산 속도를 늦춰서, 무차별 공격에 대비가 가능한 암호화 알고리즘 ↩

8. 데이터를 고정된 길이의 문자열로 변환하는 것 -- 데이터입력 ▶ 해시 함수 대입 ▶ 해시 값 생성 -- 함수이기 때문에 항상 같은 x에 대하여 같은 y 산출 ↩
9. 미리 계산된 해시값 목록 -- 원래였으면 이런 y값 목록을 보고 x값을 역추적 할 수 있지만, 솔트로 양념을 쳐놔서 x값이 뭐였는지 정확히 추측 불가 ↩
10. Open Source Computer Vision Library로 실시간 컴퓨터 비전을 위한 오픈소스 라이브러리이다. 이미지 처리, 객체 인식, 얼굴 감지 등 다양한 컴퓨터 비전 기능 제공 ↩
11. Efficient and Accurate Scene Text Detector로 이미지 내 텍스트 영역을 효율적으로 감지하는 알고리즘이다. 자연 장면에서 텍스트 위치를 식별하는 데 뛰어나다. 회전되거나 왜곡된 텍스트도 감지할 수 있어 신용카드, 문서, 간판 등의 텍스트 인식에 활용된다. ↩
12. Optical Character Recognition : 이미지에서 글자 추출하는 기술 ↩
13. 예시 : Google Vision API ↩
14. 룬 알고리즘으로 IBM의 피터 룬이 개발. 신용카드 번호 검증에 사용되는 알고리즘으로, 1954년에 개발되었기 때문에 현재 지식재산권의 보호가 끝나 퍼블릭 도메인에 사용할 수 있다. ↩
15. 지도학습을 통해 라벨이 있는 정상 거래와 이상 거래 데이터 훈련. 이후 비지도학습으로 라벨이 없는 데이터에서 패턴을 발견하여 이상치 탐지. ↩
16. 신분증 탐지만을 위해 만들어진 딥러닝 기반 객체 탐지 모델 ↩
17. 구글이 제공하는 오픈소스 OCR엔진. git-hub에서 활용할 수 있다. ↩
18. Convolutional Neural Network : 객체 탐지를 위한 딥러닝 모델로, 이미지에서 특정 특징을 학습하는데 최적화된 딥러닝 모델. ↩