

차정민

jeongmin.cha@kaist.ac.kr

010-4854-2778

<https://github.com/jeongmincha>

AI Engineer

- 사이드 프로젝트 및 대학교 **데이터 마이닝** 연구실 / 대학원 **ML-NLP** 연구실에서 2년간 연구한 경험
- 빅 데이터를 다루는 알고리즘의 시간 복잡도를 향상시켜 **추천 시스템** 알고리즘을 완성했던 경험
- 회사 및 사이드 프로젝트에서 여러 가지 프레임워크를 사용하여 **풀 스택 개발**했던 경험
- 기술 스택
 - 사용 언어: Python, Javascript (Typescript), Java, C#
 - 데이터베이스: NoSQL (MongoDB), SQL (MySQL, PostgreSQL)
 - 연구: Tensorflow, Pandas, Scikit-Learn, NLTK, genism, 시각화 (Seaborn, Matplotlib), 추천시스템 (MyMediaLite)
 - 개발: NodeJS, Flask, Django, React Native, Ionic, iOS(Apple Watch), Android, PHP
 - 그외: 테스트(Ava, Python unittest, Gtest), 문서화(Sphinx, TypeDoc)
 - Git, Vim 사용, Linux 환경, cli 환경 및 shell scripting 사용에 익숙함.

근무 경험

- **HYPERRITHM** (암호화폐 헤지 펀드 자산 운용 회사)
퀀트 개발자, 2018.03 - 2019.02
 - 1) 거래소의 이벤트에 대응하여 특정 알고리즘을 구동하고, 2) 사람이 쉽게 읽고 쓸 수 있는 알고리즘 스크립트를 받아들이며, 3) 동시에 여러 알고리즘을 비동기적으로 구동할 수 있는 **자동화 거래 트레이딩 프레임워크**를 개발 (Python)
 - Python 데이터 및 시각화 라이브러리를 이용하여 개발된 **거래 알고리즘의 효용성 검증**
 - 여러 거래 봇에서 공통적으로 사용되는 NodeJS **모듈**을 만들고, **자동 테스트 환경** 도입
- **Elice** (프로그래밍 교육 서비스 회사)
연구 및 개발 인턴, 2017.01 - 2017.04
 - 학생들의 성적과 제출 및 구동 등 서비스 내 이벤트의 상관관계를 파악하는 연구 수행
 - 모바일 앱 네비게이션과 사용성에 대한 로직 설계 / React Native 기반으로 모바일 앱 개발
- **Softpub** (eBook 문제집 서비스 회사)
개발 인턴, 2016.07 - 2016.08
 - 학생들의 모의고사 시험 결과를 기반으로 성적표를 만들어주는 Python 스크립트 개발

연구 경험

- Elicast: Embedding Interactive Exercises in Instructional Programming Screencasts (L@S 2018)
2017.09 - 2017.12 @ KAIST UILab
 - 논문에서 제안하는 상호작용이 가능한 스크린캐스트 형식의 프로그래밍 교육 플랫폼의 효용성을 검증하기 위해 실험을 설계하였고, 학생들의 설문 결과를 종합하여 분석하여 논문 작성
- An Imputation Method Using Directly Connected Neighbors in a Trust Network for Recommendation
2015.01 - 2015.10 @ Hanyang University DAKE Lab
 - 신뢰네트워크 기반 추천 시스템 개발. 텍스트 형태로 약 500GB 에 해당하는 신뢰 네트워크 데이터를 사용하여 실험을 수행하고, 시간 복잡도 향상시켜본 경험

학력

- 한국과학기술원 (KAIST) | 석사, 전산학부 (School of Computing) | 2017.03 - 2019.08
 - GPA: 3.88 / 4.30
- 한양대학교 | 학사, 컴퓨터전공 (Computer Science and Engineering) | 2012.03 - 2017.02
 - GPA: 4.08 / 4.50

추가 사항

- 자료구조 수업 조교 및 온/오프라인 프로그래밍 강사 등 약 2 년 간의 강의 경험 (2016 - 2017)
- 네이버 D2 스타트업 팩토리에서 카이스트 캠퍼스 파트너로 활동 (2017.09 - 2018.04)
- Software Maestro 프로그램 6 기 연수 (2015.07 - 2016.06)
- 국가 이공계 장학생 선정

- ※ 아래 문서는 총 5 개의 개인 프로젝트(3 개의 연구 프로젝트, 2 개의 개발 프로젝트) 들을 포함합니다.
- ※ 프로젝트는 최신 순으로 배치하였습니다.
- ※ 각 프로젝트의 (1) 설명 (2) 역할 및 담당 (3) 발생 문제 및 해결 방법 (4) 성과 위주로 작성하였습니다.
- ※ 작성한 프로젝트의 보조 결과물 (논문, 보고서, 슬라이드, 시각적인 결과물 등) 과 그 외 진행했던 프로젝트는 <https://jeongmincha.github.io/projects/ko/> 에서 확인하실 수 있습니다.

1. Best Phrases for Successful Crowdfunding Projects

2017.03 - 2017.06

- 킥스타터 데이터를 활용하여 크라우드 펀딩에서 성공적인 모금을 하기 위해서는 필요한 구, 절이 무엇인지를 NLP 기법을 통해서 파악하는 프로젝트
- 크라우드 펀딩을 성공적으로 마무리했다는 것을 1) 모금한 돈의 액수 (pledged amount), 2) 모금자의 수 (the number of backers), 3) 모금을 달성한 퍼센티지 (percentage of funds) 로 정의
- 해당 크라우드 펀딩에서 쓰인 글을 n-gram 벡터로 만들어서 feature 로 사용했고, 이를 포함한 여러 feature 들을 기반으로 앞서 정의한 성공의 지표들을 예측해내는 penalized linear regression 모델을 사용함. 이 모델을 통해 성공의 지표를 높게 예측해내는 n-gram 단어 벡터를 찾아냄.

Table 2: Top-20 phrases list. We sorted the phrases in descending order of coefficients.

pledged amount	the number of backers	percentage of funds
experimentation	experimentation	more funding
could see	less expensive	we have chosen
developers	could see	pictured
shower	difference for	pledged will
time	developers	holidays
compatible with	clicking	lists
cooler	was one	we are fully
help us	support at	forever
resistant	backer	greeting
your phone	we can afford	of several
color	behavior	backers at
less expensive	floating	terrible
display	that being	universal
help us	color	good as
the meaning	resistant	we can afford
bunch	to fruition	choose the
floating	benefits of	regardless
waterproof	please note that	to you in
data	labs	will ship
timeline	pledge for	set

Table 3: Bottom-20 phrases list. We sorted the phrases in ascending order of coefficients.

pledged amount	the number of backers	percentage of funds
tags	hesitate to	leaf
flash	was one of	rest assured
hesitate to	an alternative	card
an alternative	tags	attention to detail
font	flash	in the works
ready	bunch of	edge
touch	wrap	phoenix
devices	font	regardless of
who want to	that being said	tries to
hundreds of	as a result	went on to
wireless	be sure to	the calendar
you want	it is	for helping us
are endless	touch	autographed
cord	you want	the future of
duo	is included	appreciation for
youll	standing	to pick
charging	we look forward	of my life
high risk	every backer	this reward
address	are endless	drum
controller	brick	waiting for

주요 역할 및 담당

- 3 명이 참여한 팀 프로젝트
- 본인은 킥스타터 데이터 크롤링 / penalized linear regression 모델 구현 / 해당 모델을 평가 (evaluation) 하는 업무
- 나머지 2 명은 모델에 사용할 feature engineering, data cleaning, 크라우드 펀딩의 설명 글을 n-gram 데이터로 뽑아내는 업무를 맡음.

발생 문제 및 해결 방법

- 매우 특별한 고유 명사가 결과값으로 도출되는 경우가 많았음. 아마도 특정 클라우드 펀딩이 크게 성공한 탓에 여러 클라우드 펀딩 글에서 자주 언급되었기 때문으로 사료됨.
 - 더 큰 데이터를 사용하는 것으로 해결함. 기존에는 약 4000 개의 클라우드 펀딩 데이터를 활용하였으나, 50000 개의 클라우드 펀딩 데이터를 활용.
 - 전체 데이터셋에서 특정 횟수 이상은 쓰일 때만 n-gram 에 포함시키도록 제한하였는데, 이 threshold 값을 크게 높여 일반적으로 자주 쓰이는 명사만 결과로 나오도록 수정함.

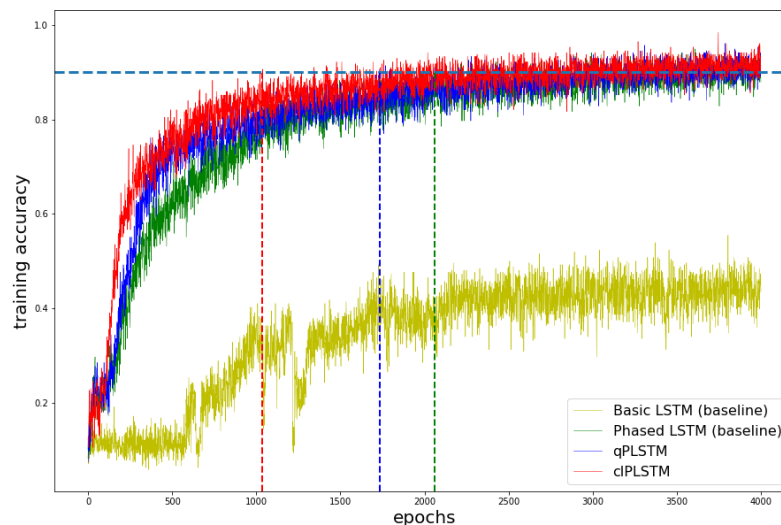
성과

- 킥스타터 데이터를 크롤링 (연구에 쓰일 데이터를 직접 모음)
- 큰 크기의 텍스트 데이터 셋 전체를 정제하고, n-gram 모델을 사용해봄

2. Phased LSTM with modified time gate function

2017.03 - 2017.06

- 베이스라인인 PhasedLSTM 에서 사용하는 time gate 함수를 변조해보면서 N-MNIST 데이터셋 (MNIST 셋과 다름)을 쓰는 기존 실험에서 더 큰 성능을 보이는 모델을 찾아봄.
- PhasedLSTM 은 LSTM 모델에서 cell 앞에 시간 (t) 에 종속적인 time gate 함수를 추가하여 긴 시퀀스의 데이터를 더 효율적으로 처리하는 모델인데, 주로 다양한 센서가 비동기적으로 데이터를 주입하는 긴 시계열 데이터를 처리하는 데 쓰임.
- time gate 함수의 open phase (cell 을 업데이트하는 순간) 를 선형에서 이차 함수로 변경해보고 (qPLSTM), 해당 함수의 close phase (cell 을 업데이트하지 않는 순간) 를 leaky function 에서 constant function 으로 변경함. (ciPLSTM)
- 그 결과 최종적으로 수렴 정확도는 기존과 큰 변화 없었으나, 더 빠른 수렴을 보여주었음.



현
<ul style="list-style-type: none"> 총 3 명이 담당하였고, 본인이 qPLSTM 모델 코드 구현을 수행 나머지 인원은 N-MNIST 데이터 정제 및 ciPLSTM 모델 구현을 수행
성과
<ul style="list-style-type: none"> 기존 논문 모델의 코드를 뜯어보면서 핵심 코드 부분을 수정해서 추가 실험을 해보는 경험을 가짐

3. An Imputation Method Using Directly Connected Neighbors in a Trust Network for Recommendation 2015.01 - 2015.10
<ul style="list-style-type: none"> 한양대학교 데이터 마이닝 연구실에서 협업 필터링 추천 시스템 성능 향상을 연구 주제로 해서 산학 프로젝트를 수행. 이 프로젝트를 통해 신뢰 네트워크에서 직접 연결된 이웃들을 활용한 대치 방법(평점 행렬을 채우는 방법)을 제안함. 기존의 대치 방법은 많은 수의 이웃들이 평점을 남긴 아이템들에 대해서만 평점 값을 예측하여 대치하였음. 하지만 제안한 방법에서는 직접 연결된 이웃에 대해서는 단 한 명만이 평가한 아이템에 대해서라도 평점 값을 예측하여 대치함.
주요 역할 및 담당
<ul style="list-style-type: none"> 본 논문의 제안 방법의 알고리즘을 C# 및 MATLAB 코드로 작성하고, 실험 결과를 정리하여 논문을 쓰는 데 반영함. 실제로 실험을 주도하고 설계하는 것은 전부 본인이 하였으나, 박사과정 선배님의 연구 지도가 있었음.
발생 문제 및 해결 방법
<ul style="list-style-type: none"> 신뢰 네트워크에서 연결된 다른 유저들이 많이 평가한 아이템들에 대해서 평점을 예측하여 평점 행렬을 채우는 (imputation) 것이 실험 방법의 핵심. 실험을 위해서 사용하는 데이터가 굉장히 커서 메모리 문제가 나는 경우가 있었음. <ul style="list-style-type: none"> 신뢰 네트워크에서 특정 거리 안에 도달할 수 있는 유저들을 neighbor로 설정하고, prediction 해야 할 아이템을 저장하고 관리하는 중에 Out Of Memory 예외 처리가 많이 발생했고, 이를 해결하기 위해서, 각 원소마다 1비트를 사용하는 Bit Array를 사용하여 해결하였음. 몇몇 기능들(특정 사용자가 특정 아이템에 대해서 평가했는지? 몇점으로 평가했는지? 특정 사용자의 평균 평점 등)을 $O(1)$ 시간 안에 작동하도록 구현하여야 했음. <ul style="list-style-type: none"> 평점 행렬을 저장할 때, 이차원 배열이 아닌, Dictionary의 배열과 HashSet의 배열로 잡아서, 평가 여부와 평점 정보들을 동시에 관리하였음. 또, 사용자들의 평균 평점을 미리 계산.

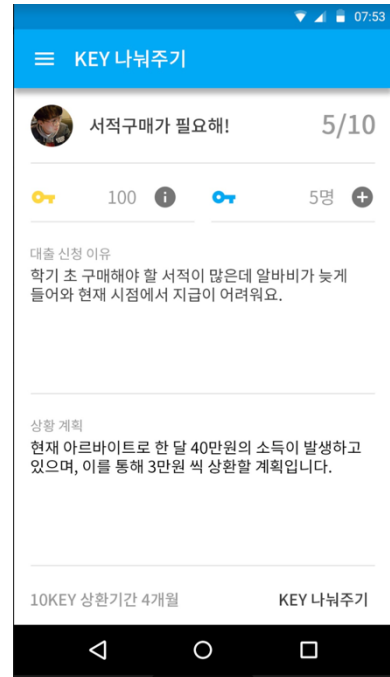
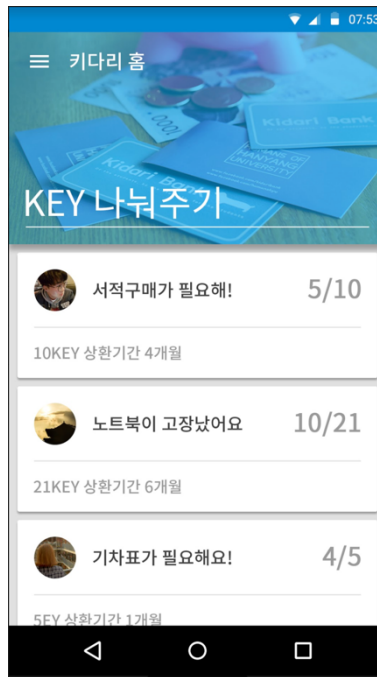
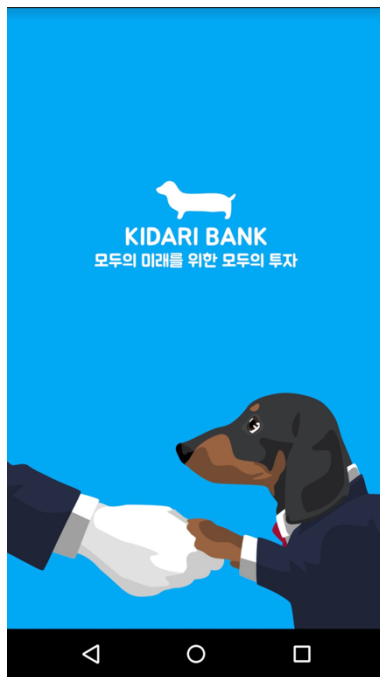
성과

- 빅데이터를 다룰 때 발생하는 메모리, 수행 속도 문제를 해결하기 위해서 세밀한 부분에서 퍼포먼스를 고려하면서 구현
- 이전까지는 특정 기술이나 언어를 이용해서 필요한 기능을 구현하는 데에 그쳤다면, 이 프로젝트를 하면서 자료구조, 메서드를 시간 공간 복잡도 측면에서 고민을 해가며 설계해야 했음. C#에서 제공하는 Collections 클래스들의 메서드들의 성능, 메모리 비용 등을 실험한 결과들을 많이 참고하면서 실험을 수행

4. 키다리 은행 클라이언트 및 서버 개발

2016.06 - 2016.12

- 대학교 학생들을 위한 은행 협동 조합인 '키다리 은행' 아웃소싱 개발 해준 프로젝트. 앱으로 대출 신청을 할 수 있고, 대출을 해 줄 사람들은 어떤 대출 신청들이 올라와 있는지를 확인할 수 있어야 함. 본인의 키다리 은행 내 보관된 금액과 기존 기록들을 확인할 수 있어야 함. 다만 실제로 돈이 오고 가는 일은 기존의 조합원들이 수행하고, 클라이언트 앱은 계좌 관리 및 대출 정보 확인 등의 목적을 수행함.



주요 역할 및 담당

- 혼자서 외주를 수주받아 진행하였음.
- 아이오닉 프레임워크를 이용하여 1) 키다리 은행 프론트엔드 앱을 개발하고, 2) 앱 구동을 위한 API 서버, 그리고 3) 관리자 페이지 를 구현함.

발생 문제 및 해결 방법

- 비록 인간이 개입하는 일에 보조적인 역할을 하는 서비스였지만, 그래도 돈이 오고 가는 일이라 잘못 동작할 경우 민감한 문제가 발생할 수 있었음.
 - 특정 버튼을 누르는 등의 동작을 할 때 중간에 문제가 발생하여 수행이 실패하더라도 나중에 기록을 보고 맞춰볼 수 있도록 로그 기록을 설계하였음. 또한, 이렇게 실패한 경우에도 문제가 없도록 반드시 상태가 롤백되도록 예외 처리를 함.
- 푸쉬 기능을 설계할 때, 한 유저가 여러 장치를 사용할 수 있는 상황을 고려하지 못하여 가장 마지막으로 로그인한 장치에 대해서만 푸쉬가 알람이 가는 문제가 발생.
 - 데이터베이스 테이블을 추가하여 각 유저별 여러 가지 디바이스 아이디를 기록하도록 구현하고, 푸쉬 기능에서 그 유저가 가지고 있는 모든 (푸쉬가 허락된) 디바이스에 보내도록 구현

성과

- 처음으로 프론트부터 서버까지 혼자 개발해본 경험 (풀 스택 개발 경험)
- 메시지 푸쉬 서버를 개발한 경험

5. Waterful (애플 워치 응용프로그램) - 수분 섭취 패턴 예측

2015.09 - 2015.11

- 애플워치 응용프로그램 개발을 주제로 시작한 프로젝트, 사용자의 수분 섭취를 유도하는 응용 프로그램. 애플 워치 인터페이스를 통해 기존의 유사 어플리케이션보다 더 손쉽고 간편하게 사용하는 것이 가능



주요 역할 및 담당

- 총 세 명이 참여하였음.

<ul style="list-style-type: none"> • 본인은 사용자의 건강 정보와 iOS의 헬스킷 프레임워크를 사용하여 사용자에게 유의미한 알림을 줄 수 있도록 수분 섭취 패턴을 예측하는 알고리즘을 구현하여 프론트엔드 앱에 붙이는 작업을 진행하였고, • 나머지 두 명이 앱 디자인 기획 및 iOS 컴패니언 앱 및 애플워치 앱의 프론트 엔드 작업을 수행
발생 문제 및 해결 방법
<ul style="list-style-type: none"> • 사용자의 자연스러운 수분 섭취를 위해서는 백그라운드 모드에서의 로직이 많이 필요했는데, 헬스킷 프레임워크의 쿼리는 백그라운드 모드에서 사용할 수 없었음. <ul style="list-style-type: none"> ○ 헬스킷 쿼리 중 HKObserverQuery를 사용하면 백그라운드에서도 헬스킷 데이터 샘플이 추가되는 것을 감지할 수 있음. 이 쿼리를 사용하여 사용자의 걸음과 물 섭취 기록에 대한 샘플이 추가되는 것을 수집하여 수집되는 '시간'을 기준으로 시계열 패턴을 수집하고, 사용자가 일어난 시간을 예측하거나 물 마실 시간을 패턴하는 알고리즘에 사용할 수 있었음. • 사용자의 수분 섭취 패턴을 적절하게 예측하는 알고리즘을 구현해야 함 <ul style="list-style-type: none"> ○ 기존의 연구 중 K-Means 클러스터링과 AprioriAll 알고리즘을 활용하여 재고 경향의 시계열 패턴을 분석하는 방법에 대한 연구 논문이 있어, 그 방법을 참고하여 물 섭취 패턴을 예측하는 알고리즘을 구현함.
성과
<ul style="list-style-type: none"> • Swift 언어 및 iOS의 헬스킷 프레임워크, 애플워치 개발을 사용해보는 경험 • 기존 서비스와의 차별을 위해 알림을 유의미한 시간에 전달할 수 있는 알고리즘을 고민해본 경험