

보드 게임을 통한 단어 군집화 인공지능

누가 기침소리를 내었는가 팀

김태희 변성진 임성근 차승호

전반부 발표: 차승호

후반부 발표: 임성근

9조

목차

1. 과제 개요

2. 과제 해결 과정

3. 과제 세부 진행 내용 및 시연

4. 목표 달성 검토 여부

5. 기대효과 및 의의

6. 한계점 및 개선사항

7. 역할분담

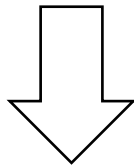
8. 피드백

1.1 과제 개요 – 배경

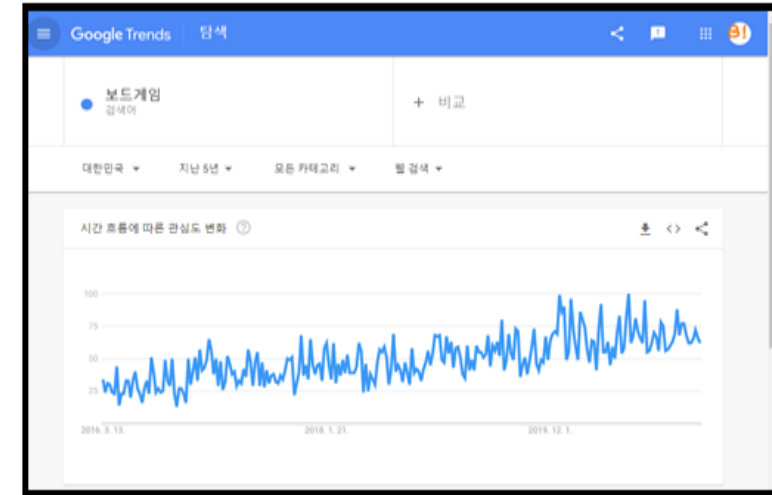
보드게임 관심도 증가

앱 게임 사용 수 증가

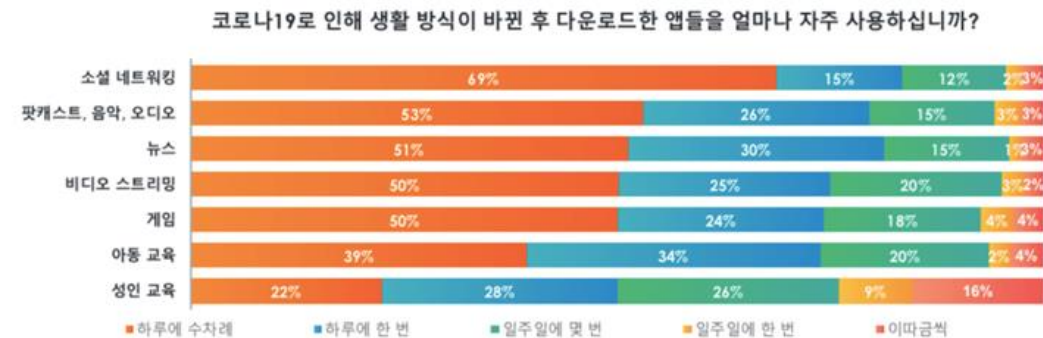
앱으로 보드게임을 하며 얻을 수 있는 데이터가 존재한다면



사람들이 게임을 즐기면서 데이터를 생성



Google trends 보드게임 검색 동향



1.1 과제 개요 - 배경



다양한 **그림이 그려진 카드**를 뽑은 후,
그에 대한 **키워드**를 듣고
진열된 카드 중에서
해당 **카드가 무엇인지 맞추는** 게임

그림 한 장을 뽑아 키워드 제시

키워드 - 데이터

맞추는 과정에서 관념적으로 공감되는 단어 생성
(사전적 정의 -> 게임에서 패배)

1.2 과제 개요 - 필요성



소개	인식기술-시각지능	인식기술-언어지능	인식기술 - 기타	추론기술	표현기술
수어 데이터셋	웹니스 대학 스크립트 데이터셋	한국어 대학 데이터셋			
트위터에서 수집 및 정제한 대화 시나리오	어린이 음성 데이터셋	VRM 화행 데이터			
한국어 감정 정보가 포함된 연속적 대화 데이터셋	한국어 감정 정보가 포함된 단발성 대화 데이터셋	인공지능 음리 연구를 위한 비정형 텍스트 데이터셋			
심리상담을 위한 멀티모달 데이터셋	감정 분류를 위한 대화 음성 데이터셋				



과학기술정보통신부

NIA 한국지능정보사회진흥원



nipa 정보통신산업진흥원



ETRI 한국전자통신연구원

언어와 관련된
데이터 셋이 많지 않음
(규모 x)

일일이 라벨화 된 데이터
(확장성 x)

네이버 영화리뷰 데이터



의도 학습 데이터셋
3i4k

1.2 과제 개요 – 필요성

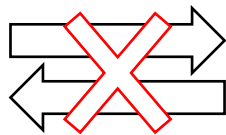
- <Sonnet 18> 윌리엄 셰익스피어

"Shall I compare thee to a summer's day?"

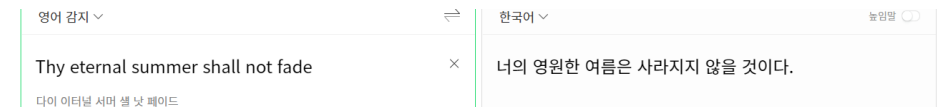
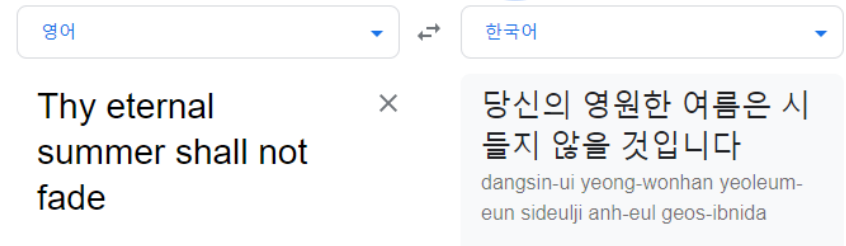
...(중략)

Thy eternal summer shall not fade"

영원한 여름



사랑



여름

[명사] 한 해의 네 철 가운데 둘째 철. 봄과 가을 사이이며, 낮이 길고 더운 계절로, 달로는 6~8월, 절기(節氣)로는 입하부터 입추 전까지를 이른다.

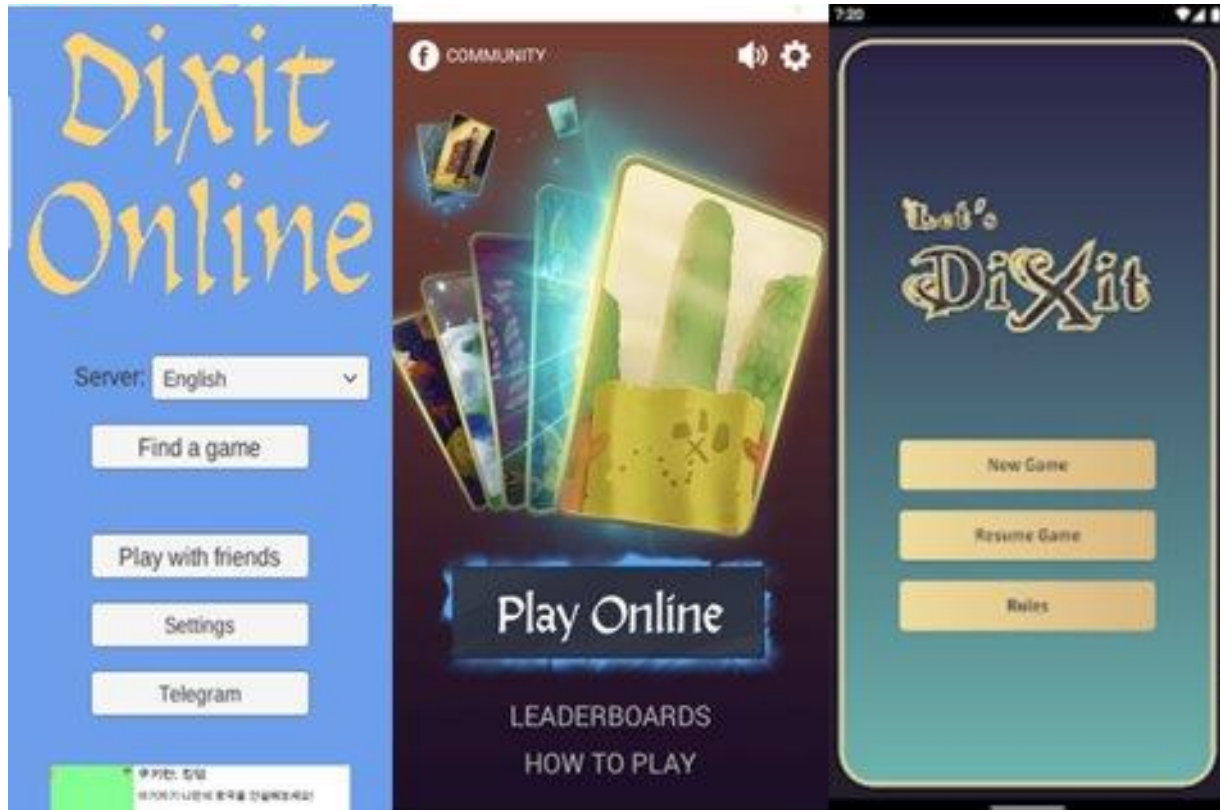
[유의어] 구하, 하계⁵, 여름철

영원하다

[형용사] 어떤 현상, 형편, 모양 따위가 끝없이 이어지는 상태이다. 또는 시간을 초월하여 변하지 아니하는 상태이다.

[유의어] 끝없다, 무한하다, 불후하다

1.3 과제 개요 – 유사 서비스



1. 한국어 지원을 하지 않는다.

2. UI

3. 단어 및 문장의 제한이 없다.

2.1 과제 해결 과정 - 간트 차트



2.2 사용 프레임 워크



Android
Studio



2.3 통신규약

HTTP

JSON 형식

병렬적인 프로그래밍

방만들기	앱->서버	<pre>{ "code": "REQUEST_ROOM_CREATE", "user_num": "사용자고유번호(누군지 알수있게)", "room_name": "방이름", "room_user": "6" //가능한 유저 수? default로 6? }</pre>
	서버->앱	<pre>{ "code": "SUCCESS_ROOM_CREATE" //"FAILED_ROOM_CREATE"시엔 나머지 다 "0" "user_num": "사용자고유번호", //누가요청했는지 "room_name": "방이름", "room_no": "방 고유번호", "room_user": "6", //층 몇명까지 들어갈 수 있는지 "user_list": [{ "user_name": "유저아이디", "user_num": "사용자고유번호", "user_score": "사용자 점수", "is_ready": "1" //방장은 없애고 모든사람 준비가 시작 }, { "user_name": "유저아이디", "user_num": "사용자고유번호", "user_score": "사용자 점수", "is_ready": "0" }] }</pre>

2.3 머신러닝 및 인공지능 구현



임베딩

1. CPU 기반 환경에서 CPU intensive한 트레이닝 사용 (GPU 사용 x)

2. 트레이닝에 사용되지 않은 단어에 대한 임베딩 가능

3. Common Crawl 과 Wikipedia의 문서로 사전 학습된 모델 제공



분류모델

1. 의사 결정 트리 기반의 gradient boosting 프레임워크

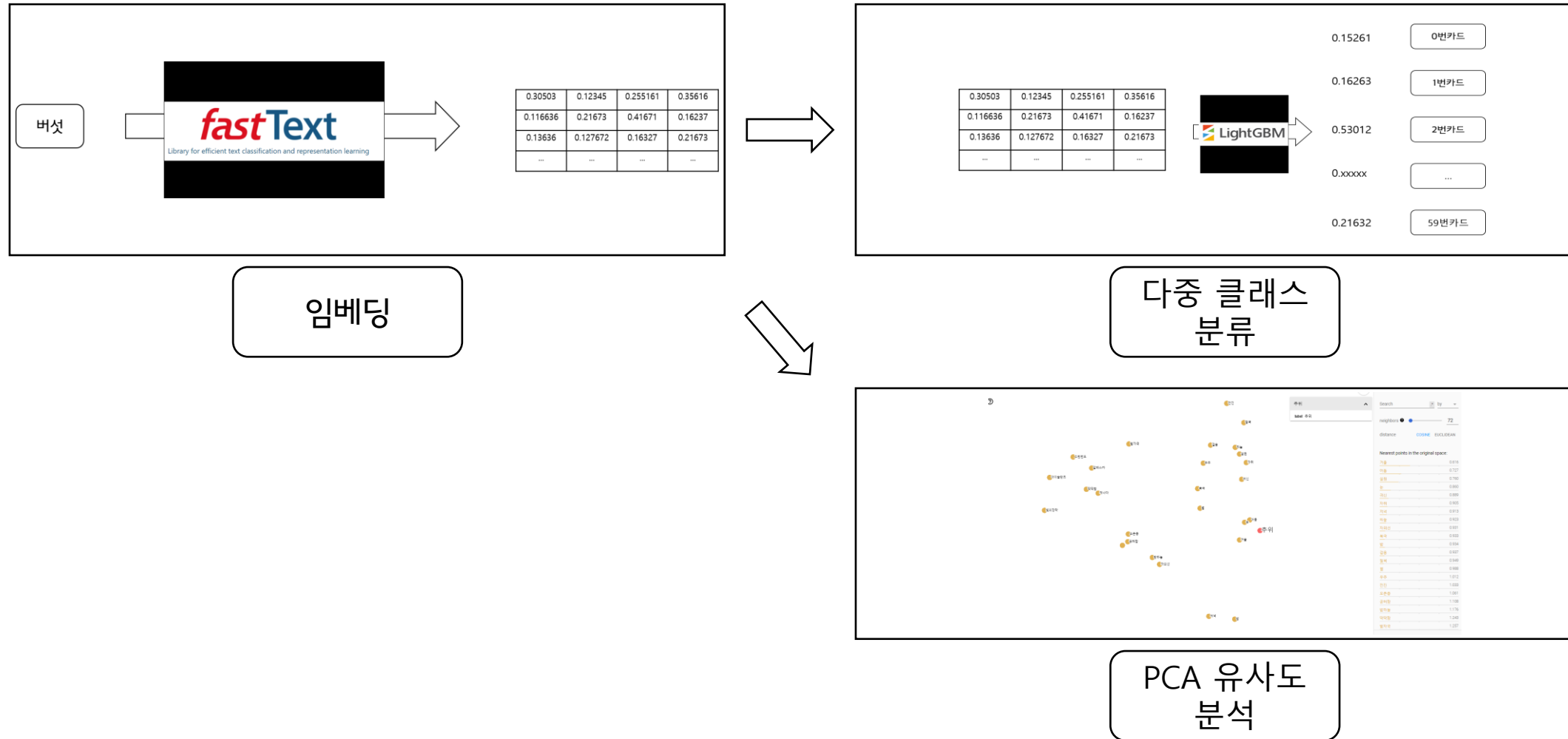


분류모델



개발

2.3 머신러닝 및 인공지능 구현 과정



2.3 머신러닝 및 인공지능 구현

LIGHTGBM

<https://drive.google.com/file/d/1TxomwVr9yUpLzKCJbuxCXntTHqZ7TYhy/view?usp=sharing>

```
numofcard=len(card_data)
numofvec=300 #embedding vector 크기
cardnumli=np.zeros(numofcard) #label로쓸거 (multinomial classification)
keywordli=np.zeros((numofcard,numofvec))

i=0
for cardno,keyword,usernum in tqdm(card_data):
    cardnumli[i]=cardno
    curvec=pretmod.wv[keyword]
    for j in range(300):
        keywordli[i][j]=curvec[j]
    i+=1

keywordli.shape

weightli=[1]*numofcard #각 데이터의 가중치를 줄 수있어
weightnpli=np.array(weightli)
```

```
parammult={'num_leaves':100,#100 #multinomial classiffication
            'min_data_in_leaf':30,
            'objective':'multiclass',
            'max_depth':11 , #6 10
            'learning_rate':0.02 ,#'learning_rate':0.02 ,
            'boosting':"gbdt",
            "feature_fraction":0.8, #0.7522
            "bagging_freq":1,
            "bagging_fraction":0.7083,
            "bagging_seed":11,
            "metric":"multi_logloss",
            "num_class":60,
            "random_state":133
}
```

임베딩된 단어

카드 번호

60개의 카드에
대한 분류

2.3 머신러닝 및 인공지능 구현



```
[ ] testword=pretmod.wv["솔로"]
test_x=np.zeros((1,300))
for idx in range(300):
    test_x[0][idx]=testword[idx]
pred_y_list=[]
for model in models:
    pred_y=model.predict(test_x)
    pred_y_list.append(pred_y.reshape(-1,1))

pred_ensemble=np.mean(pred_y_list,axis=0)
print(pred_ensemble.argmax())
print(pred_ensemble[pred_ensemble.argmax()])
print(len(pred_ensemble))
```

4
[0.06657503]
60

```
def AITries(keyword, answer, cardli): # AI에게 keyword가 전달되고, answer를 입력받아 AI의 정답과 비교
    curword = keyword
    votenum = -1
    votenum_li = []
    voterand = random.randrange(0, len(cardli)) # 0번 AI는 무조건 random으로 뽑는 애
    votenum = cardli[voterand]
    votenum_li.append(votenum)

    if PTmodel.has_index_for(curword): # index있으면
        testword = PTmodel[curword]
        test_x = np.zeros((1, 300))
        pred_y_list = []
        for idx in range(300):
            test_x[0][idx] = testword[idx]
        for modelli in model_list:
            for models in modelli:
                pred_y = models.predict(test_x)
                pred_y_list.append(pred_y.reshape(-1, 1))
        pred_ensemble = np.mean(pred_y_list, axis=0)
        maxidx = 0
        maxval = pred_ensemble[cardli[0]]
        for j, val in enumerate(cardli):
            if (pred_ensemble[val] > maxval):
                maxval = pred_ensemble[val]
                maxidx = j
        votenum = cardli[maxidx]
        votenum_li.append(votenum)
```

출제자



'ET' 키워드로
출제된 카드



어떤 카드가 'ET' 인가?



사람1



사람2



사람3



머신러닝
결과물

```
for i in AILI:
    print(str(i.number) + "의 현재 점수 ", str(i.tries) + "시도 중 ", str(i.correct) + "정답 ", str(i.wrong) + "틀림")
```

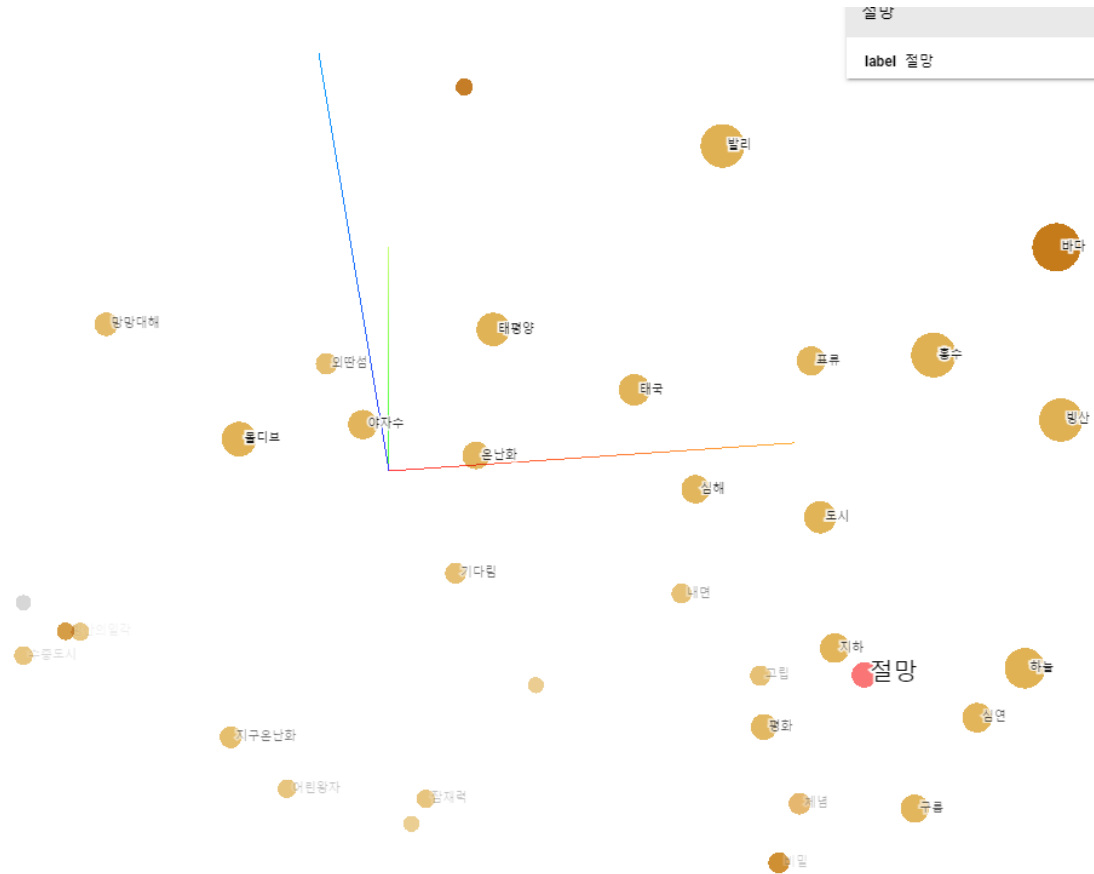
0의	현재 점수	9시도	2정답	7틀림
1의	현재 점수	9시도	1정답	8틀림
2의	현재 점수	9시도	4정답	5틀림



0의	현재 점수	34시도	9정답	25틀림
1의	현재 점수	34시도	10정답	24틀림
2의	현재 점수	34시도	18정답	16틀림

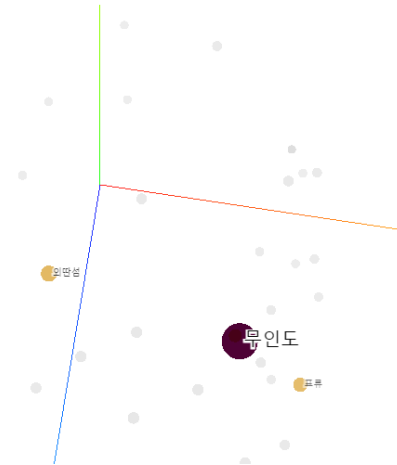
- 0: 랜덤하게 선택
- 1: 5월 17일까지의 데이터로 분류
- 2: 5월 24일까지의 데이터로 분류

2.3 머신러닝 및 인공지능 구현



Nearest points in the original space:

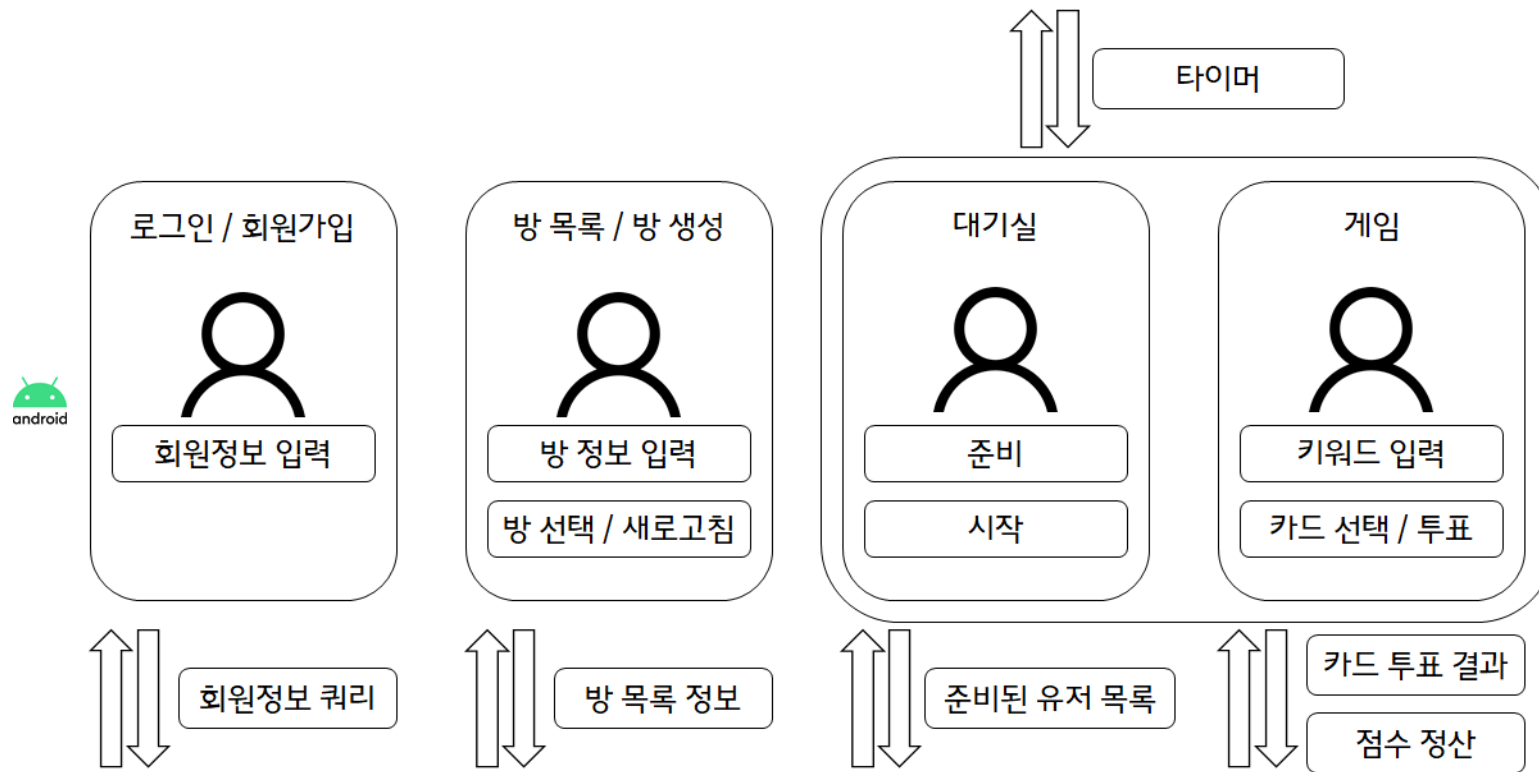
제념	0.575
심연	0.608
고립	0.688
구름	0.734
하늘	0.741
평화	0.748
홍수	0.778
빙산	0.803
외로움	0.828
지하	0.844
표류	0.846
발리	0.855
바다	0.864
바다	0.864
태국	0.872
비밀	0.873
비밀	0.873
내면	0.880
기다림	0.926
도시	0.932
쓸쓸함	0.972



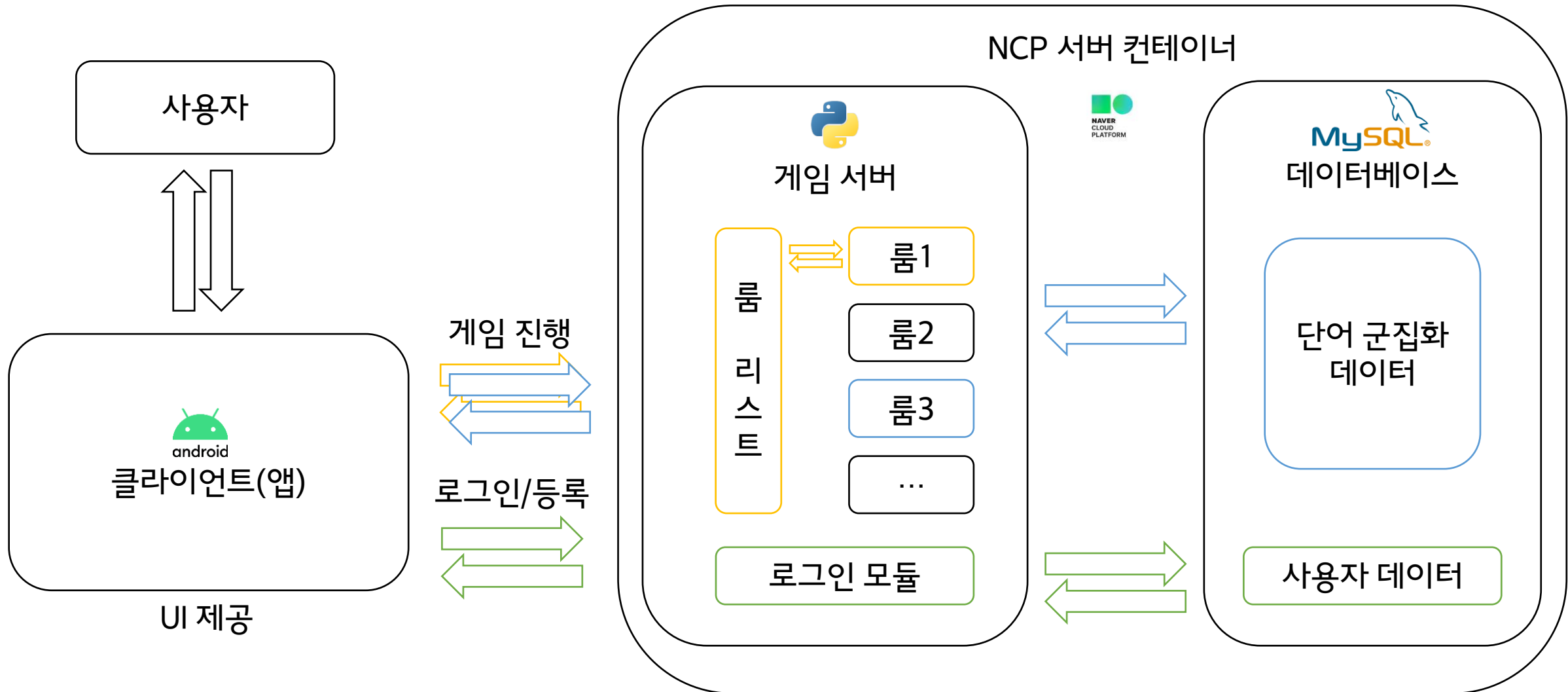
Nearest points in the original space:

무인도	0.000
무인도	0.000
외딴섬	0.860
표류	0.913
심해	0.948

3.1 세부 내용 - 앱



3.2 세부 내용 - 서버



3.2 세부 내용 - 앱, 서버 시연



<https://youtu.be/8SHprKkejwE>

3.3 세부 내용 – 데이터베이스(1)

로그인 정보

유저 번호 (서버 부여)

아이디

비밀번호 (해시)

이메일

전적 정보

수행한 경기 수

현재까지 얻은 점수

현재까지 정답 횟수

키워드를 제시한 횟수

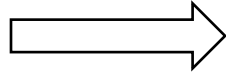
제시한 키워드로 다른 사람들이 내 카드를 맞춘 횟수

다른 사람의 키워드로 내 카드를 선택한 횟수

usernum	user_id	user_pw	user_score	user_email	user_name	totalcnt	answer	similarcnt	totaltellercnt	tellerw
76	39076	f1d1275e41309cd5c262a5e8c66196c4977c2ed...	0	0	39076	0	0	0	0	0
77	39077	4496f74e0f5afaa7f212755b3c3a806355f121d8...	74	0	39077	35	8	17	10	6
78	39078	2cb4a18f3e4c7bd3ec8d79e1bade9f563d38149...	43	0	39078	31	5	11	12	1
79	39079	b507ccd25172b63816b03fec29e324bea1654f8...	40	0	39079	31	3	13	10	2
80	39080	431bf5b656745970d95541720af5f410fa3a207...	50	0	39080	22	10	4	5	4
81	39081	bd8f994314a41ed7882e5a54abc1a52c8a6313b...	14	0	39081	8	3	0	3	1
82	39082	f38542f0d2d181fda770936a0a8c9118d5b6af3c...	3	0	39082	5	0	1	1	0

3.3 세부 내용 - 데이터베이스(2)

cardno	cardkeyword	usernum
50	동물	47
34	영광	47
37	밤샘	60
13	승강장	60
5	생명	47
42	전국	47
4	비행	47
59	표현	46
47	도둑	46
12	연못	47
7	우산	47
41	더위사냥	46



37번



13번



5번

카드-단어 정보

카드 번호

키워드

해당 키워드를 통해서 해당 카드를 선택한 유저 번호

3.4 세부 내용 - 머신러닝 / 인공지능

분류 문제

```
testword=pretmod.wv["결혼"]
test_x=np.zeros((1,300))
for idx in range(300):
    test_x[0][idx]=testword[idx]
pred_y_list=[]
for model in models:
    pred_y=model.predict(test_x)
    pred_y_list.append(pred_y.reshape(-1,1))

pred_ensemble=np.mean(pred_y_list,axis=0)
print(pred_ensemble.argmax())
print(pred_ensemble[pred_ensemble.argmax()])
print(len(pred_ensemble))
for idx,val in enumerate(pred_ensemble):
    print(idx,val)
```

```
35
[0.28936923]
60
0 [0.01251779]
1 [0.00436254]
2 [0.01511651]
3 [0.01294444]
4 [0.00983967]
5 [0.00854305]
6 [0.01483298]
7 [0.01426337]
```

```
mysql> select * from card where cardkeyword="결혼 ";
+-----+-----+-----+-----+
| idcard | cardno | cardkeyword | usernum |
+-----+-----+-----+-----+
| 2758 | 35 | 결혼 | 47 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

35번



['결혼'] 과 1번째로 유사한 카드

58번



['결혼'] 과 2번째로 유사한 카드

36번



['결혼'] 과 3번째로 유사한 카드

3.4 세부 내용 - 머신러닝 / 인공지능

분류 문제

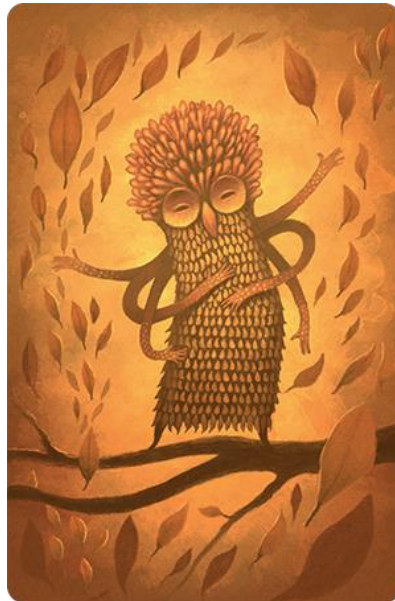
```
testword=pretmod.wv["고독"]
test_x=np.zeros((1,300))
for idx in range(300):
    test_x[0][idx]=testword[idx]
pred_y_list=[]
for model in models:
    pred_y=model.predict(test_x)
    pred_y_list.append(pred_y.reshape(-1,1))

pred_ensemble=np.mean(pred_y_list,axis=0)
print(pred_ensemble.argmax())
print(pred_ensemble[pred_ensemble.argmax()])
print(len(pred_ensemble))
for idx,val in enumerate(pred_ensemble):
    print(idx,val)
```

```
18
[0.31630714]
60
0 [0.01562532]
1 [0.00509611]
2 [0.01392002]
3 [0.00969022]
4 [0.00794584]
5 [0.01354461]
6 [0.0161753]
7 [0.0095912]
```

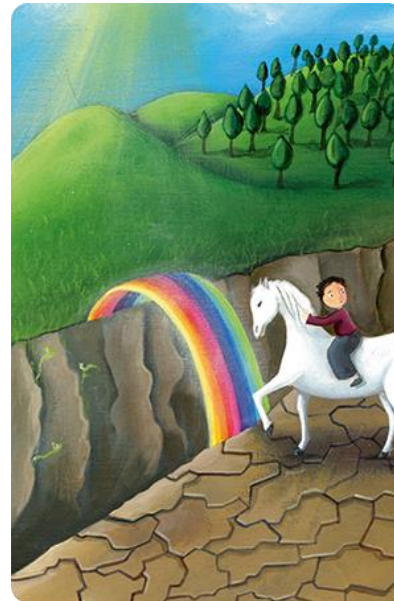
```
mysql> select * from card where cardkeyword= '고독';
+-----+-----+-----+-----+
| idcard | cardno | cardkeyword | usernum |
+-----+-----+-----+-----+
| 667 | 18 | 고독 | 60 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

18번



['고독'] 과 1번째로 유사한 카드

42번



['고독'] 과 2번째로 유사한 카드

37번



['고독'] 과 3번째로 유사한 카드

3.4 세부 내용 - 머신러닝 / 인공지능

분류 문제

```
testword=pretmod.wv["추격"]
test_x=np.zeros((1,300))
for idx in range(300):
    test_x[0][idx]=testword[idx]
pred_y_list=[]
for model in models:
    pred_y=model.predict(test_x)
    pred_y_list.append(pred_y.reshape(-1,1))

pred_ensemble=np.mean(pred_y_list,axis=0)
print(pred_ensemble.argmax())
print(pred_ensemble[pred_ensemble.argmax()])
print(len(pred_ensemble))
for idx,val in enumerate(pred_ensemble):
    print(idx,val)
```

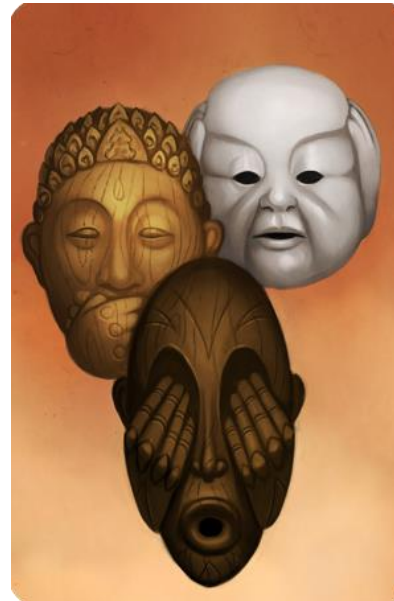
```
0
[0.03760358]
60
0 [0.03760358]
1 [0.00533402]
2 [0.0176804]
3 [0.01041069]
4 [0.01261302]
5 [0.01304248]
6 [0.02316681]
7 [0.0178593]
```

0번



['추격'] 과 1번째로 유사한 카드

59번



['추격'] 과 2번째로 유사한 카드

54번



['추격'] 과 3번째로 유사한 카드

```
mysql> select * from card where cardkeyword="추격 ";
Empty set (0.00 sec)
```

3.4 세부 내용 - 머신러닝 / 인공지능

유의어 사전

조종	:	와	가장	유사한	단어는	:	조작	:	0.463646
49번 카드	:					:			
편지	:	와	가장	유사한	단어는	:	서신	:	0.516297
불	:	와	가장	유사한	단어는	:	재	:	0.296095
구멍	:	와	가장	유사한	단어는	:	주름	:	0.338894
단서	:	와	가장	유사한	단어는	:	비밀	:	0.315389
빈칸	:	와	가장	유사한	단어는	:	영어	:	0.341462
영어	:	와	가장	유사한	단어는	:	빈칸	:	0.341462
두루마리	:	와	가장	유사한	단어는	:	양피지	:	0.33715
양피지	:	와	가장	유사한	단어는	:	두루마리	:	0.33715
소실	:	와	가장	유사한	단어는	:	기억	:	0.333228
편지	:	와	가장	유사한	단어는	:	서신	:	0.516297
흉터	:	와	가장	유사한	단어는	:	상처	:	0.54304
아픔	:	와	가장	유사한	단어는	:	상처	:	0.507839
상처	:	와	가장	유사한	단어는	:	흉터	:	0.54304
흔적	:	와	가장	유사한	단어는	:	흉터	:	0.475349
글씨	:	와	가장	유사한	단어는	:	필기체	:	0.368779
글	:	와	가장	유사한	단어는	:	편지	:	0.387397
종이	:	와	가장	유사한	단어는	:	양피지	:	0.328532
눈물	:	와	가장	유사한	단어는	:	아픔	:	0.429177
세익스피어	:	와	가장	유사한	단어는	:	영어	:	0.250437
주름	:	와	가장	유사한	단어는	:	흉터	:	0.505386
세월	:	와	가장	유사한	단어는	:	이별	:	0.355929
기억	:	와	가장	유사한	단어는	:	흔적	:	0.400268
취급	:	와	가장	유사한	단어는	:	주문	:	0.289572
주의사항	:	와	가장	유사한	단어는	:	취급	:	0.249652
이별	:	와	가장	유사한	단어는	:	아픔	:	0.439032
작별	:	와	가장	유사한	단어는	:	이별	:	0.415376
마법	:	와	가장	유사한	단어는	:	주문	:	0.338309
서신	:	와	가장	유사한	단어는	:	편지	:	0.516297
편지	:	와	가장	유사한	단어는	:	서신	:	0.516297
깃털	:	와	가장	유사한	단어는	:	주름	:	0.350603
재	:	와	가장	유사한	단어는	:	불	:	0.296095
비밀	:	와	가장	유사한	단어는	:	단서	:	0.315389
서약	:	와	가장	유사한	단어는	:	약속	:	0.370601
종이	:	와	가장	유사한	단어는	:	양피지	:	0.328532

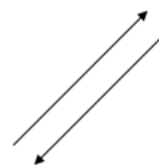
3.4 세부 내용 - 머신러닝 / 인공지능

유의어 사전

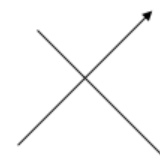
유의어	유의어	코사인 유사도
무한	영원	0.35857
탈주	탈옥	0.39504
질투	탐욕	0.43246
행글라이더	비행	0.38989
종말	파괴	0.37367
비밀	진실	0.39957
횡재	행운	0.52958
공돌이	공대생	0.56955

유의어	유의어	코사인 유사도
지니	요슬램프	0.34596
꿈	소원	0.35783
오페라의유령	뮤지컬	0.42471
기괴	놀람	0.34167
유혹	덧	0.52725
인내	끈기	0.49919
정보관	도서관	0.36885
공허함	쓸쓸함	0.56786

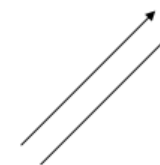
```
f=open(PATH+"0522_sim.txt",mode="wt",encoding='utf-8') #데이터를 저장할 파일
curwordli=[] #현재 카드번호의 단어 리스트
worli=[]
simli=[] #유사도가 얼마나 되는지에 대한 측정
i=0
for wordidx in range(len(a.keys())):
    wordli=a[wordidx].split(' ') #24 카드로 가지고 있는 단어를 mysql에서 받아와 a에 a['0']은 0번카드의형태로 저장
    curwordli=[]
    worli=[]
    simli=[]
    for outeridx in range(len(wordli)):
        temp=[]
        wordtemp=[]
        curword=wordli[outeridx]
        for inneridx, val in enumerate(wordli):
            if(outeridx==inneridx):
                continue
            temp.append(pretmod.wv.similarity(curword,wordli[inneridx]))
            wordtemp.append(wordli[inneridx])
        simli.append(temp)
        worli.append(wordtemp)
        curwordli.append(curword)
    b=np.array(simli)
    f.write(str(wordidx)+"번 카드\n")
    for idx, val in enumerate(simli):
        v=np.argmax(val)
        f.write(curwordli[idx]+":와 가장 유사한 단어는:"+ str(worli[idx][v])+ " :("+str(val[v])+")\n")
f.close()
```



코사인 유사도: -1



코사인 유사도: 0



코사인 유사도: 1

4. 목표 달성 여부 검토

1. 한국어로 플레이 할 수 있는 딕싯 게임 어플리케이션 구현

2. 사용자들의 플레이를 통한 데이터 확보

3. 확보된 관념적 유의어 데이터 셋을 이용한 단어 군집화 AI

4. 목표 달성 여부 검토

딕싯 보드 게임의 기본 규칙이 잘 작동 하였는가?

어플리케이션과 서버의 연결이 안정적인가?

UI 가 게임을 플레이 하기에 적절한가?

단어 군집화를 위한 데이터를 축적할 수 있는가?

인공지능 플레이어가 딕싯 게임을 플레이 할 수 있는가?

4. 목표 달성 여부 검토

정성적 목표

사용자가 딥싱 게임을 플레이 하는데 원활한 인터페이스를 제공한다.

AI 플레이어와 플레이 하였을 때, 누가 AI 플레이어인지 맞추지 못하게 한다. - AI의 승률

사람들이 동의할만한 관념적 동의어 데이터 셋을 구축한다. -설문조사

정량적 목표

관념적 동의어에 관한 데이터 수를 최소 2000개 이상 확보한다. - 데이터의 수

각 단어에 대한 관념적 동의어를 2개 이상 확보한다. -PCA 그래프 및 정리 파일

4. 목표 달성 여부 검토

유의어	유의어	코사인 유사도
무한	영원	0.35857
질투	탐욕	0.43246
종말	파괴	0.37367
비밀	진실	0.39957
횡재	행운	0.52958
기괴	놀람	0.34167
유혹	덧	0.52725
공허함	쓸쓸함	0.56786
염탐	악행	0.44652
엔지니어	공대생	0.33398

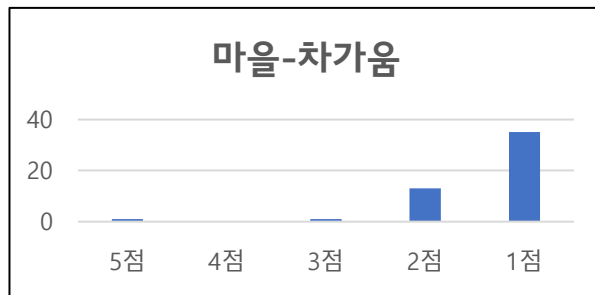
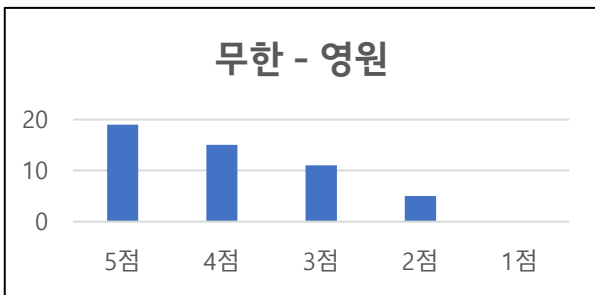
설문 유의어 목록

단어1	단어2	코사인 유사도
산소통	다재다능	0.04955
등불	폐쇄공포증	0.03612
자유	줄무늬	-0.02062
공원	최후	-0.01391
성실	비행기	-0.06557
마을	차가움	0.05624
황혼	주황색	-0.04890
위협	피라냐	0.08113
주객전도	역전	0.00296
요정	미스터리	0.07410

설문 일반 단어 목록 (유사도 ~ 0)

단어1	단어2	코사인 유사도
아낌없이주는나무	소멸	-0.06538
무인도	평화	0.03786
인어	차가움	0.10859
기술	z세대	0.01681
커튼	준비중	0.05324

4. 목표 달성 여부 검토



50명의
설문 조사

종합설계 프로젝트 설문조사

유사도 선택

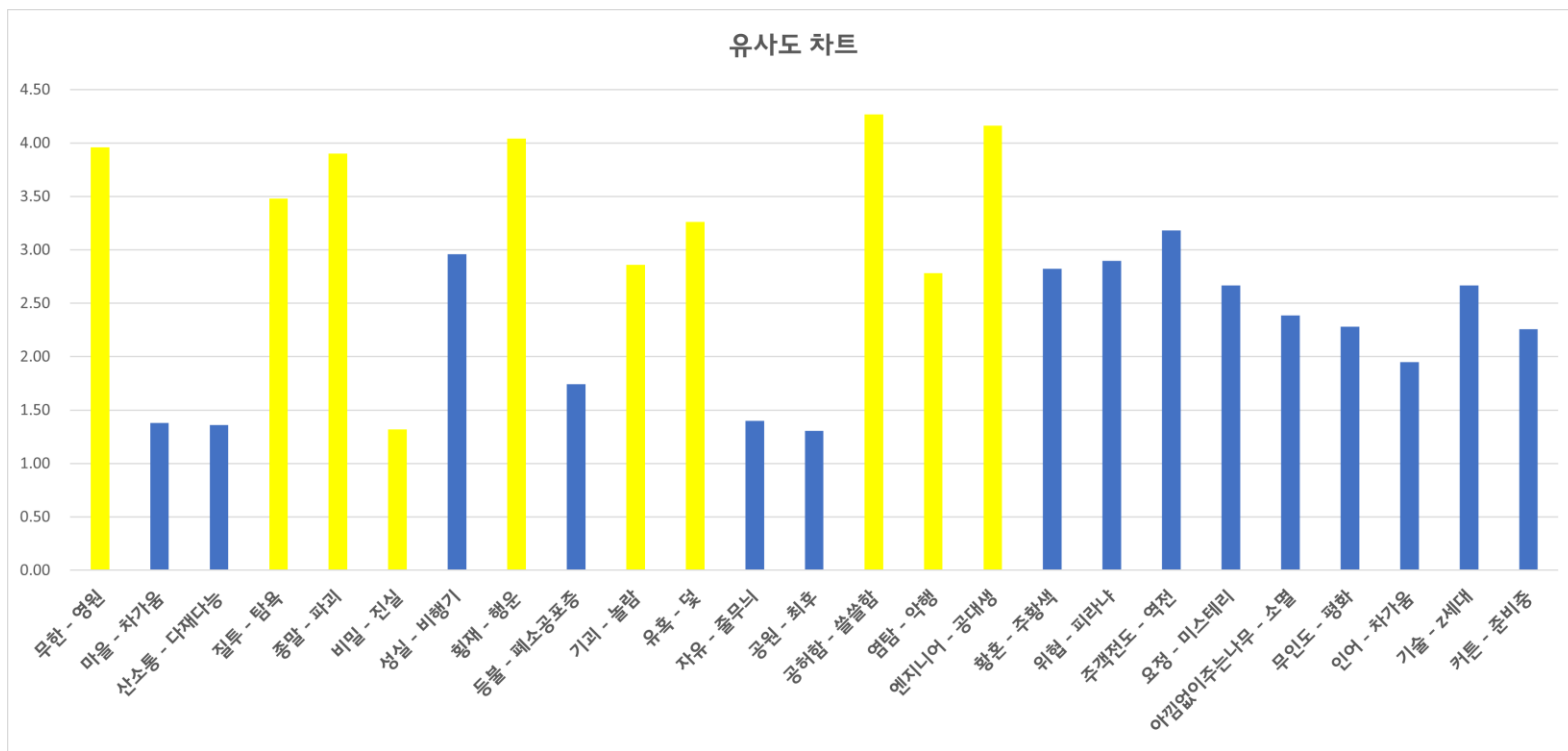
다음 두 단어의 유사도가 어느정도로 판단되는지 선택해 주십시오

다음 단어가 얼마나 유사해 보이십니까?

	아무 관계 없 어 보인다.	조금 관계 있 어 보인다.	관계 있어 보 인다.	많이 유사해 보인다.	매우 유사해 보인다.
무한 - 영원	○	○	○	○	○
마음 - 차가움	○	○	○	○	○
산소통 - 다재 다능	○	○	○	○	○
	1	2	3	4	5

[illegible]

4. 목표 달성 여부 검토



유의어 평균 점수



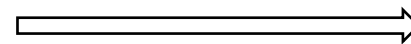
3.40

일반 단어 평균 점수



2.21

매우 유사함



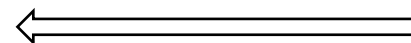
1

2

3

4

5



매우 관계 없음

4. 목표 달성 여부 검토

단어

유의어

유의어

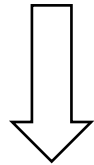
	A	B	C	D	F	H	K	M	N	O
2561		가면	:	무도회	0.339682	:	탈	:	0.266295	
2562		마스크	:	얼굴	0.31792	:	대머리	:	0.253873	
2563		미궁	:	거미	0.312607	:	기괴	:	0.310765	
2564		눈가림	:	입막음	0.359561	:	눈속임	:	0.354304	
2565		눈속임	:	눈가림	0.354304	:	입막음	:	0.315558	
2566		놀람	:	기괴	0.341671	:	감정	:	0.323381	
2567		나무	:	목재	0.517875	:	거미	:	0.311934	
2568		목재	:	나무	0.517875	:	규제	:	0.303455	
2569		가면	:	무도회	0.339682	:	탈	:	0.266295	
2570		부족	:	통제	0.295249	:	인종	:	0.277808	
2571		기괴	:	놀람	0.341671	:	미궁	:	0.310765	
2572		입막음	:	납치	0.370282	:	눈가림	:	0.359561	
2573		표현	:	감정	0.445235	:	감정	:	0.445235	
2574		자유	:	언론	0.39004	:	억압	:	0.366268	
2575		억압	:	통제	0.434032	:	규제	:	0.385922	
2576		언론	:	종교	0.423626	:	자유	:	0.39004	
2577		통제	:	규제	0.461884	:	억압	:	0.434032	
2578		규제	:	통제	0.461884	:	언론	:	0.38844	
2579		가면	:	무도회	0.339682	:	탈	:	0.266295	
2580		미스터리	:	납치	0.269227	:	귀신	:	0.241903	
2581		숨김	:	억압	0.264871	:	미궁	:	0.257203	
2582		감정	:	표현	0.445235	:	억압	:	0.377868	
2583		포커페이스	:	눈속임	0.273005	:	입막음	:	0.257386	
2584		레이디가가	:	눈코입	0.318586	:	오페라의유령	:	0.292185	
2585		탈	:	가면	0.266295	:	가면	:	0.266295	
2586		하회탈	:	눈코입	0.427245	:	장발	:	0.31552	
2587		웃음	:	거미	0.314489	:	놀람	:	0.310846	
2588		거짓	:	언론	0.318735	:	납치	:	0.290983	

5. 기대효과 및 의의

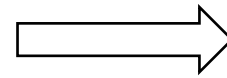
보드게임 관심도 증가

앱 게임 사용 수 증가

앱으로 보드게임을 하며 얻을 수 있는 데이터가 존재한다면



사람들이 게임을 즐기면서
데이터를 생성



소개	인식기술-시각지능	인식기술-언어지능	인식기술-기타	추론기술	표현기술
수어 데이터셋	블리스 대학 스크립트 데이터셋	한국어 대학 데이터셋			
트위터에서 수집 및 정제한 대화 시나리오	어린이 음성 데이터셋	VRM 화행 데이터			
한국어 감정 정보가 포함된 연속적 대화 데이터셋	한국어 감정 정보가 포함된 단발성 대화 데이터셋	인공지능 윤리 연구를 위한 비정형 텍스트 데이터셋			
심리상담을 위한 말리모달 데이터셋	감정 분류를 위한 대화 음성 데이터셋				

그림 6 AIHUB에서 제공하는 데이터셋

5. 기대효과 및 의의



"커튼"와 가장 유사한 단어:

- ('암막', 0.66120404)
- ('커텐', 0.60673934)
- ('등화관제용', 0.58943897)
- ('베딩류', 0.51190877)
- ('커튼이나', 0.50724632)
- ('래그', 0.48197046)
- ('침구', 0.47728249)
- ('스ミノ工', 0.47645444)
- ('샤워용', 0.4759177)
- ('차광', 0.46611035)

"커튼"와 가장 유사한 단어:

- ('거울', 0.33527502)
- ('무대', 0.32759482)
- ('계단', 0.32181406)
- ('공연', 0.3170414)
- ('바람', 0.2909171)
- ('염탐', 0.26427102)
- ('진실', 0.24143198)
- ('교복', 0.23666285)
- ('가림막', 0.22475463)
- ('무도회', 0.20463918)



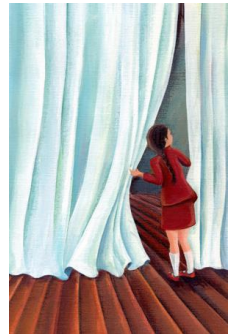
영어

한국어

Thy eternal
summer shall not
fade

×

당신의 영원한 여름은 시
들지 않을 것입니다
dangsin-ui yeong-wonhan yeoleum-
eun sideulji anh-eul geos-ibnida



5. 기대효과 및 의의

card (5).png

leisure

전체

이미지

지도

쇼핑

더보기

설정

도구

검색결과 약 2개 (0.66초) << Add Grepper Answer (a)

이미지 크기: 1242 x 2208

다른 크기의 해당 이미지를 찾을 수 없습니다.

관련 있을 가능성이 높은 검색어: **leisure**

Show 2 More Grepper Results

<https://dictionary.cambridge.org> > 사전 > 영어 > leisure

영어로 **leisure**의 뜻 - Cambridge Dictionary

The town lacks **leisure** facilities such as a swimming pool or squash courts. 추가 예. Swimming pools usually come under '**leisure** centres' in the phone directory. I ...

<https://livewirewarrington.co.uk> > leisure

Leisure | LiveWire Warrington

Gym; Fitness Classes; Online Booking; **Leisure** FAQs; Swimming; Tennis; Our Locations; Membership options; Personal Training; Fit to Tackle BootCamp ...

유사한 이미지

여가

여가 또는 레저는 직업상의 일이나 필수적인 가사 활동 외에 소비하는 시간이다. 먹기, 자기, 일하러 가기, 사업하기, 수업에 출석하기, 숙제하기, 집안일과 같은 의무적인 활동 전후에 남는 자유로운 시간이다. 올바른 여가를 체험하려면, 다음의 세 가지 기준을 만족해야 한다. 위키백과

관련 검색어

10개 이상 항목 더보기

게임

오락

사업

건강

대체 텍스트

시각 장애가 있거나 시력이 낮은 사용자에게 이 개체 및 해당 컨텍스트를 어떻게 설명하시겠어요?

- 주제에 대한 자세한 내용

- 설정

- 동작 또는 상호 작용

- 관련 있는 기타 정보

(1-2개의 자세한 문장 권장)

커튼, 바닥이(가) 표시된 사진

자동 생성된 설명

설명 생성

☐

장식으로 표시

대상 단어				1번째 유의어				2번째 유의어			
커튼	:	거울	0.33527	:	무대	:	0.32757				
소녀	:	교복	0.322227	:	성장	:	0.277309				
무대	:	공연	0.494917	:	계단	:	0.364642				
비밀	:	진실	0.399574	:	염탐	:	0.317134				

6. 한계점 및 개선사항

한계점

1. 문학 등에 쓰일 수 있을 것이라고 생각한 관념적 유의어 학습에 데이터가 많이 필요하다.

2. 코사인 유사도가 수치적으로 높지 않다.

3. 동음 이의어 분리의 어려움



솔로 가수?

커플이 아닌
외로운 솔로?

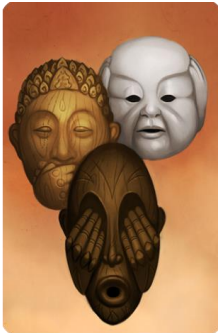
6. 한계점 및 개선사항

개선사항

결혼과
어울리는 카드



['결혼'] 과 1번째로
유사한 단어



['결혼'] 과 2번째로
유사한 단어



['결혼'] 과 3번째로
유사한 단어



['결혼'] 과 4번째로
유사한 단어

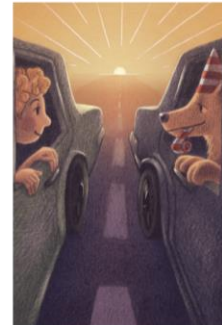
5월 17일 모델
(500여개의 데이터)



['결혼'] 과 1번째로
유사한 단어



['결혼'] 과 2번째로
유사한 단어



['결혼'] 과 3번째로
유사한 단어

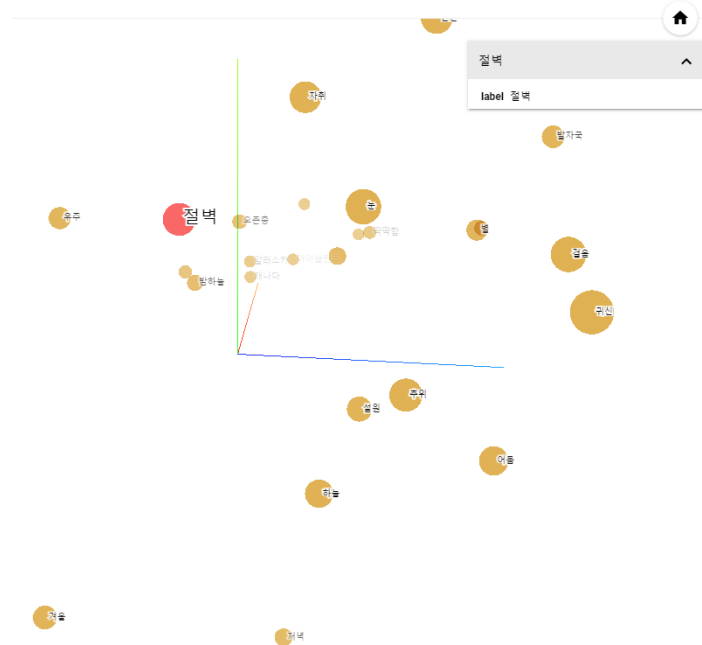
5월 24일 모델
(2300여개의 데이터)

6. 한계점 및 개선사항

개선사항

['절벽']에
대한 단어

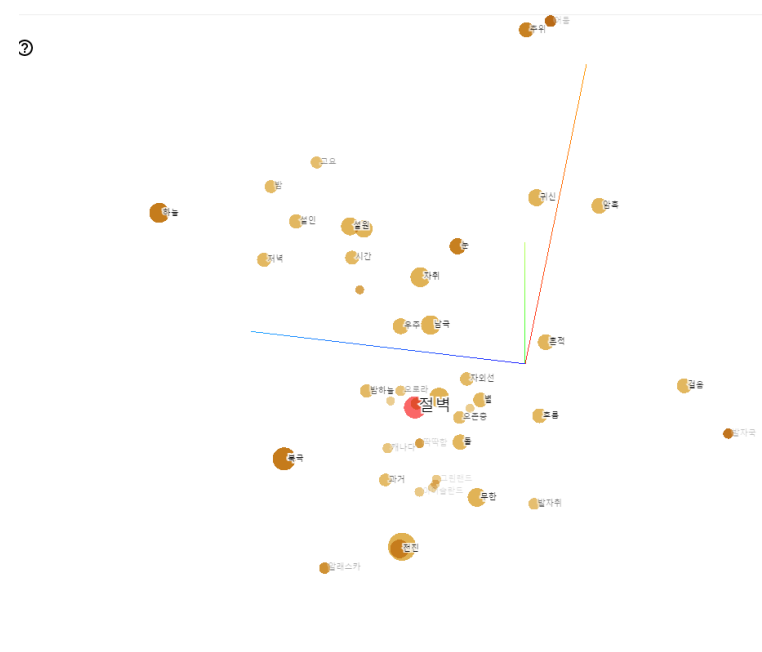
imension: 300 | Selected 28 points



설원
하늘
복귀

5월 17일 모델
(500여개의 데이터)
해당 카드
데이터 :28개

Points: 56 | Dimension: 300 | Selected 56 points



암석
남극
설원

5월 24일 모델
(2300여개의 데이터)
해당 카드
데이터 :56개

6. 한계점 및 개선사항

개선사항

1. 데이터의 부족으로 관념적 유의어가 적음

2. 문학적 DB의 사용을 통해 여러번 나타나는 단어 묶음을 사전 학습한 모델을 사용

3. 상업적 이용을 지원하기 위한 메뉴 개설 - 데이터 라벨링 도구로서의 활용성

7. 역할 분담

역할 (담당파트)	성 명	계획 발표 역할	중간 발표 역할	최종 발표 역할
팀장 (서버)	변성진	sequence diagram 작성 (로그인, 회원가입, 방 새로고침, 방 만들기, 방 나가기), 제안서 작성 및 ppt 제작 (시나리오, 기능적요소, 배경 및 필요성), 서버 객체 분석, 간트차트 제작, 계획 발표(전반부)	서버 제작 (로그인, 회원가입, DB연동, 타이머 함수, 방 접속, 나가기), 네이버 클라우드 서비스 서버 세팅, 중간 발표 PPT 제작 (서버 진행상황, 미래 계획, 역할 분담)	결과 보고서 및 ppt 제작 (표지, 인공지능 구현 및 해결 방안 과정, 한계점, 개선사항, 업 무 분담, 소요비용), multinomial classification 모델 제작 및 서버 이식, 단어 데이터 확보, 유사도 분석 데이터 제작 및 그래프화, 회의록 문서 관리, 영수증 처리,
팀원 (앱)	차승호	sequence diagram 작성 (방 접속, 준비완료, 게임시작), 제안서 작성 및 ppt 제작 (유사 서비스, 과제 목표, 기대 효과, 프로젝트 관리), 간트차트 제작, 계획 발표(후반부)	어플리케이션 제작 (카드배분, 텔러지정, 투표), 중간 발표 PPT 제작 (과제 개요 및 동기 요약)	결과 보고서 및 ppt 제작 (요약, 서버와 앱 구현 및 해결 방안 과정, 과제 추진 배경 및 목적), 앱 배포 단계로 수정, 군집화 된 텍스트 데이터 분류, 단어 데이터 확보, 최종 발표
팀원 (서버)	김태희	sequence diagram 작성 (게임 종료, 단어 데이터 업데이트, 방 보존), 서버 객체 분석, 제안서 작성 및 ppt 제작 (서버, 클라이언트 기능 및 구조), 간트 차트 제작, 서버/클라이언트 구조도 제작	서버 제작 (게임 룸 구성, 룸 생성, 룸 조회, 카드 배분, 키워드 분배, 투표), 중간 발표 PPT 제작 (계획발표 피드백 진행상황)	결과 보고서 및 ppt 제작 (과제 배경, 유사 서비스, 데이터 형식, 백엔드, 데이터베이스, 기대효과), multinomial classification 모델 제작 및 서버 이식, 단어 데이터 확보
팀원 (앱)	임성근	sequence diagram 작성 (teller 주제입력, 카드선정, 투표), 제안서 작성 및 ppt 제작 (데이터베이스, 인공지능 부분 기술), 간트 차트 제작, 어플리케이션 UI 스케치	어플리케이션 제작 (로그인, 화면, 방 목록, 방 생성, 방 대기, 레이아웃, 액티비티 구성, 타이머 스레드), 중간 발표 PPT 제작 (앱 진행사항 설명)	결과 보고서 및 ppt 제작 (sequence diagram, 통신 규약, 앱 구현 관련 세부 내용, 테스트 결과), 앱 배포 단계로 수정, 군집화 된 텍스트 데이터 분류, 단어 데이터 확보 최종 발표

8. 피드백

역할 (담당파트)	성명	피드백
팀장 (서버)	변성진	유저와 상호작용이 많고 실시간으로 게임을 진행해야 하다 보니 프로그램의 구현 난이도가 많이 높아졌다. 데이터를 모으고 분류 모델을 사용했을 때나 유사도 분석해 유의어를 모았을 때 성능이 좋았고, 단순히 기성 사전만으로는 알기 어려운 유의어들을 많이 모을 수 있어 만족스러운 결과물이라고 느껴졌다.
팀원 (앱)	차승호	여러 유저가 함께 쓰는 게임 어플리케이션의 특성상 생각보다 시간이 오래 걸렸고 기존에 해봤던 것보다 어려움이 많이 있었다. 그래도 서버와의 통신이 잘 연결된 결과물을 보았을 때 뿌듯했고, 인공지능 부분을 맡지는 않았지만 유의어의 결과물이 많이 공감되어 실생활에 충분히 쓸 수 있을 것 같았다.
팀원 (서버)	김태희	프로젝트를 통해서 간단하게나마 온라인 서비스를 개발하는 경험과 자연어처리를 적용하는 경험을 함께 할 수 있었던 뜻 깊은 경험이었다. 돌이켜보면 이런 프로그램을 개발하는 과정이 결코 쉽지는 않았지만, 팀원들의 도움 덕분에 훌륭하게 프로젝트를 마무리할 수 있었다. 더하여 이렇게 보드게임을 활용해 언어 데이터를 수집하는 방법이 프로젝트를 넘어 현실에서도 충분히 효과적으로 활용될 수 있겠다는 생각도 함께 들었다.
팀원 (앱)	임성근	프로젝트 초기에 예상했던 것보다 애플리케이션 구현 난이도가 높았으나, 팀원들과의 협업덕에 만족스러운 결과물을 도출하여 원활하게 플레이할 수 있었다. 또한, 플레이데이터를 활용하여 생성된 모델이 사전적 동의어 이외의 유의어에도 높은 유사도를 보이므로 프로젝트 초기 목표를 성공적으로 달성하였다.

Q&A

들어주셔서 감사합니다.