코 알 라 유 니 브 2 기 해 커 톤

사용자 맞춤 악성 댓글 필터링

발 표 자 권 예 진

중앙대학교 공공경경

CONTENTS

중앙대학교 공공경경

팀원소개

01

박소현

데이터 전처리
모델링, PPT제작

고주형

프론트/서버 개발
서버 배포

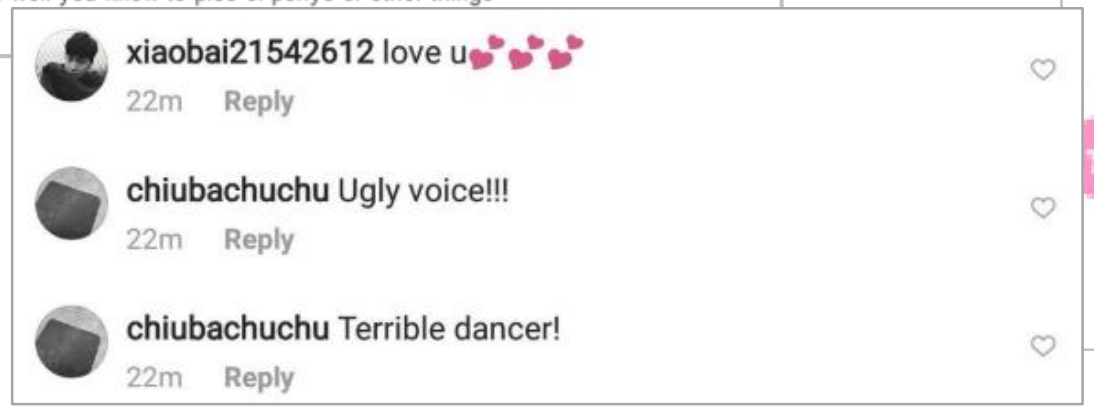
이혜지

데이터 전처리
모델링

권예진

데이터 전처리
PPT제작,발표


salsav91 11 minutes ago
So based on those statistics. All bronies are gay. You actively go looking for bronies to interact with. Meaning you are gay. Those are your statistics, not mine.. But its good to know you are gay as well too.
Reply · in reply to soren balthes
soren balthes 7 minutes ago
at lest i don't... you know to pics of ponys
Reply · in reply to salsav91
salsav91 3 minutes ago
You didn't deny being gay.. Well that"s good to admit to ones sexuality. Baby steps, baby steps.
Reply · in reply to soren balthes
soren balthes 1 minute ago
i never... well you know to pics of ponys or other things
Reply

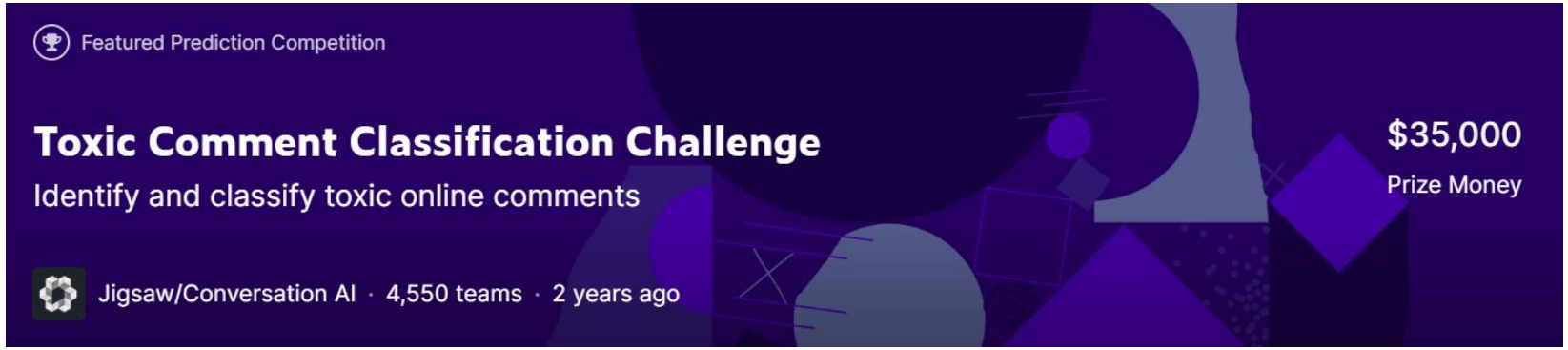

xiaobai21542612 love u
22m Reply
chiubachuchu Ugly voice!!!
22m Reply
chiubachuchu Terrible dancer!
22m Reply


@missA_suzy 연예계에서 추방되라교통사고나서 죽어버려
자세히
@missA_suzy 재수없는 인간아.
자세히
숯이
@missA_suzy
제가 죽
12:15 PM · 09 Nov 14
설리 집에서 숨진 채 발견
MBC
주는대로 돌려받는다.
나는 누구에게 사랑을주고 상처를 줬나.
나는 누구에게 사랑을 받고 상처를 받았나.
NEWS DESK
악성댓글·루머로 고통·· 우울증 호소
6분 전
entertain.naver.com
근데 진짜 이시언 보기 불편하긴 함. 근래 10년지기 친구가 나한테 하는 말꼬라지가 저러면 기분 나쁠텐데. 내가 편하면 남이 불편하고, 내가 불편하면 남이 편해요. 방성환, 기안 84 데이트편 보면 진짜 기안84 성격이 나오니깐요.
13분 전 답글 1 공감 2 비공감 0
'나혼자산다' 기안84 "이시언, 악플에 힘들다고 전화"
entertain.naver.com

“ 사용자 맞춤 악성 댓글 필터링 서비스 ”

Kaggle Dataset 이용

Online Toxic Comment Dataset



	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\r\nWhy the edits made under my use...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\r\nMore\r\nI can't make any real suggestions...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\r\n\r\nCongratulations from me as well, use ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0

데이터 전처리

다중 분류에서 이진 분류로

	id	comment_tex	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\r\nWhy the edits made under my use...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\r\nMore\r\nI can't make any real suggestions...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0
5	00025465d4725e87	"\r\n\r\nCongratulations from me as well, use ...	0	0	0	0	0	0
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1				
7	00031b1e95af7921	Your vandalism to the Matt Shirvington article...	0	0				
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0				
9	00040093b2687caa	alignment on this subject and which are contra...	0	0				

	comment_text	is_toxic
0	Explanation\r\nWhy the edits made under my use...	0
1	D'aww! He matches this background colour I'm s...	0
2	Hey man, I'm really not trying to edit war. It...	0
3	"\r\nMore\r\nI can't make any real suggestions...	0
4	You, sir, are my hero. Any chance you remember...	0
5	"\r\n\r\nCongratulations from me as well, use ...	0
6	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1
7	Your vandalism to the Matt Shirvington article...	0
8	Sorry if the word 'nonsense' was offensive to ...	0
9	alignment on this subject and which are contra...	0

데이터 전처리

오버피팅을 일으킬 수 있는 쓸모없는 단어들을 제거

Explanation₩n Why the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms. I'm retired [Doory]. 89.205.28.27

소문자로
개행/닉네임/IP주소 제거

explanation why the edits made under my username hardcore metallica fan were reverted? they weren't vandalisms. i'm retired.

데이터 전처리

단어 토큰화 및 클렌징

explanation why the edits made under my username hardcore metallica fan were reverted? they weren't vandalisms. i'm retired.

단어 토큰화

['explanation', 'why', 'the', 'edits', 'made', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'were', 'reverted', '?', 'they', 'weren't', 'vadalisms', '.', 'i'm', 'retired']

데이터 전처리

단어 토큰화 및 클렌징

['explanation', 'why', 'the', 'edits', 'made', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'were', 'reverted', '?', 'they', 'weren't', 'vadalisms', '.', 'i'm', 'retired']

줄임말 풀이

['explanation', 'why', 'the', 'edits', 'made', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'were', 'reverted', '?', 'they', 'were', 'not', 'vadalisms', '.', 'i', 'am', 'retired']

데이터 전처리

단어 토큰화 및 클렌징

['explanation', 'why', 'the', 'edits', 'made', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'were', 'reverted', '?', 'they', 'were', 'not', 'vadalisms', '.', 'i', 'am', 'retired']

Stemming으로 어근찾기

['explanation', 'why', 'the', 'edit', 'make', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'be', 'revert', '?', 'they', 'be', 'not', 'vadalisms', '.', 'i', 'be', 'retire']

데이터 전처리

단어 토큰화 및 클렌징

['explanation', 'why', 'the', 'edit', 'make', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'be', 'revert', '?', 'they', 'be', 'not', 'vadalisms', '.', 'i', 'be', 'retire']

Punctuation 제거

['explanation', 'why', 'the', 'edit', 'make', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'be', 'revert', 'they', 'be', 'not', 'vadalisms', 'i', 'be', 'retire']

데이터 전처리

단어 토큰화 및 클렌징

['explanation', 'why', 'the', 'edit', 'make', 'under', 'my', 'username', 'hardcore', 'metallica', 'fan', 'be', 'revert', 'they', 'be', 'not', 'vadalisms', 'i', 'be', 'retire']

불용어(Stopwords) 제거

['explanation', 'edit', 'make', 'username', 'hardcore', 'metallica', 'fan', 'revert', 'vadalisms', 'retire']

데이터 전처리

TF-IDF를 이용한 피처 벡터화

TF

(Term Frequency)

문서 내에 얼마나 자주 등장하는지를
나타내는 값

×

IDF

(Inverse Document Frequency)

다른 문서에서 자주 등장하는
정도의 역수

=

문서 내에 자주 등장할수록 ↑
다른 문서에서 자주 등장할수록 ↓

모델링

학습/검증 데이터로 분리 및 하이퍼 파라미터 설정

```
X_features = tfidf_vect.fit_transform(df['comment_text'])
y_labels = df['is_toxic']

X_train, X_test, y_train, y_test = train_test_split(X_features, y_labels, test_size=0.2, random_state=0)
```

1. 라벨 데이터와 피처 데이터로 분리

2. 학습/검증 데이터로 분리

```
params = {
    'num_leaves' : [32, 64],
    'max_depth' : [128, 160],
    'min_child_samples' : [60, 100],
    'subsample' : [0.8, 1],
}
```

3. 하이퍼 파라미터 설정

모델링

LightGBM 모델 생성/GridCV를 이용한 교차 검증

4. LightGBM Classifier 객체 생성

```
lgbm_clf = LGBMClassifier(n_estimators=300)
gridcv = GridSearchCV(lgbm_clf, param_grid=params)
gridcv.fit(X_train, y_train, early_stopping_rounds=30, eval_metric="auc", eval_set=[(X_train, y_train), (X_test, y_test)]
```

```
[78]    valid_0's auc: 0.964765 valid_0's binary_logloss: 0.124648      valid_1's auc: 0.949839 valid_1's binary_logloss: 0.138692
[79]    valid_0's auc: 0.964944 valid_0's binary_logloss: 0.124327      valid_1's auc: 0.949974 valid_1's binary_logloss: 0.138474
[80]    valid_0's auc: 0.965071 valid_0's binary_logloss: 0.124042      valid_1's auc: 0.95007  valid_1's binary_logloss: 0.138352
[81]    valid_0's auc: 0.965244 valid_0's binary_logloss: 0.12374       valid_1's auc: 0.950231 valid_1's binary_logloss: 0.13807
[82]    valid_0's auc: 0.965416 valid_0's binary_logloss: 0.123467      valid_1's auc: 0.950386 valid_1's binary_logloss: 0.137896
[83]    valid_0's auc: 0.965625 valid_0's binary_logloss: 0.123177      valid_1's auc: 0.950371 valid_1's binary_logloss: 0.137799
[84]    valid_0's auc: 0.965783 valid_0's binary_logloss: 0.122888      valid_1's auc: 0.950562 valid_1's binary_logloss: 0.137558
[85]    valid_0's auc: 0.966004 valid_0's binary_logloss: 0.122591      valid_1's auc: 0.95069  valid_1's binary_logloss: 0.137343
[86]    valid_0's auc: 0.966125 valid_0's binary_logloss: 0.122326      valid_1's auc: 0.950857 valid_1's binary_logloss: 0.137159
[87]    valid_0's auc: 0.966305 valid_0's binary_logloss: 0.122085      valid_1's auc: 0.951058 valid_1's binary_logloss: 0.136959
[88]    valid_0's auc: 0.966458 valid_0's binary_logloss: 0.121828      valid_1's auc: 0.951305 valid_1's binary_logloss: 0.13681
[89]    valid_0's auc: 0.966601 valid_0's binary_logloss: 0.121555      valid_1's auc: 0.951456 valid_1's binary_logloss: 0.136629
[90]    valid_0's auc: 0.966776 valid_0's binary_logloss: 0.121298      valid_1's auc: 0.951505 valid_1's binary_logloss: 0.136535
[91]    valid_0's auc: 0.966962 valid_0's binary_logloss: 0.121058      valid_1's auc: 0.951535 valid_1's binary_logloss: 0.136406
[92]    valid_0's auc: 0.967132 valid_0's binary_logloss: 0.12078       valid_1's auc: 0.951644 valid_1's binary_logloss: 0.136214
[93]    valid_0's auc: 0.967273 valid_0's binary_logloss: 0.120524      valid_1's auc: 0.951807 valid_1's binary_logloss: 0.136008
[94]    valid_0's auc: 0.967366 valid_0's binary_logloss: 0.120288      valid_1's auc: 0.951849 valid_1's binary_logloss: 0.135869
```

5. GridCV로 최적의 하이퍼 파라미터를 찾으면서 모델 학습

```
최적의 하이퍼 파라미터 모델 : {'max_depth': 128, 'min_child_samples': 60, 'num_leaves': 32, 'subsample': 0.8}
```

모델링

정확도/정밀도/재현율/F1Score 평가 지표

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

pred = gridcv.predict(X_test)
print('정확도:{0:.4f}, 정밀도:{1:.4f}, 재현율:{2:.4f}, f1:{3:.4f}'.format(accuracy_score(y_test, pred), precision_score(y
```

6. 정확도/정밀도/재현율/F1Score 평가지표 출력

```
정확도:0.9580, 정밀도:0.9002, 재현율:0.6684, f1:0.7671
```

정확도

올바르게 예측된 데이터 수/전체 데이터 수

F1 Score

정밀도와 재현율의 조화평균

정밀도

실제로 True인 데이터수/모델이 True로 예측한 데이터

재현율

모델이 True라고 인식한 데이터수/ 실제로 True인 데이터

결과

예시 문장을 모델에 넣어 예측한 결과

```
sent = "I really hate you! You bastard fuck you!!!!!!!!!"
result= preprocessing_test(sent)
input_feature = result[0]
score = result[1]
```

테스트 데이터셋 전용 전처리 함수

```
입력된 댓글 :
 I really hate you! You bastard fuck you!!!!!!!!!

악성 댓글 여부 : True
```

03

"어떻게 악성의 정도를 판단할 수 있을까?"

```
def bad_amount(comment):
    bad_words = pd.read_csv("./bad_words.csv").en_bad_words.tolist()

    # bad word counting
    count_of_word = len(comment.split())+1
    count_of_bad = sum((1 for word in bad_words if comment.count(word)>0))
    percent_of_bad = count_of_bad / count_of_word

    # sentimental analysis
    sent_score = sentAnalyzer.polarity_scores(comment)['compound']

    final_score = -sent_score + percent_of_bad

    return final_score
```

1. 욕리스트로 욕 단어 개수 카운팅

2. 전체 단어에서 욕 단어 비율을 계산

3. Vader를 통한 감정분석

- (감정분석결과) + (욕단어개수/전체단어개수)

03

“ 어떻게 악성의 정도를 판단할 수 있을까? ”

	comment_text	is_toxic	bad_amount
0	explanation edit make username hardcore metall...	0	-0.515733
1	d'aww match background colour seemingly stick ...	0	-0.361200
2	hey man really try edit war guy constantly rem...	0	0.241500
3	cannot make real suggestions improvement wonde...	0	-0.250000
4	sir hero chance remember page	0	-0.680800
5	congratulations well use tool well · talk	0	-0.796400
6	cocksucker piss around work	1	1.378300
7	vandalism matt shirvington article revert plea...	0	0.318200
8	sorry word nonsense offensive anyway intend wr...	0	0.571900
9	alignment subject contrary dulithgow	0	0.000000

서버 개발

flask로 서버 개발 및 heroku로 배포

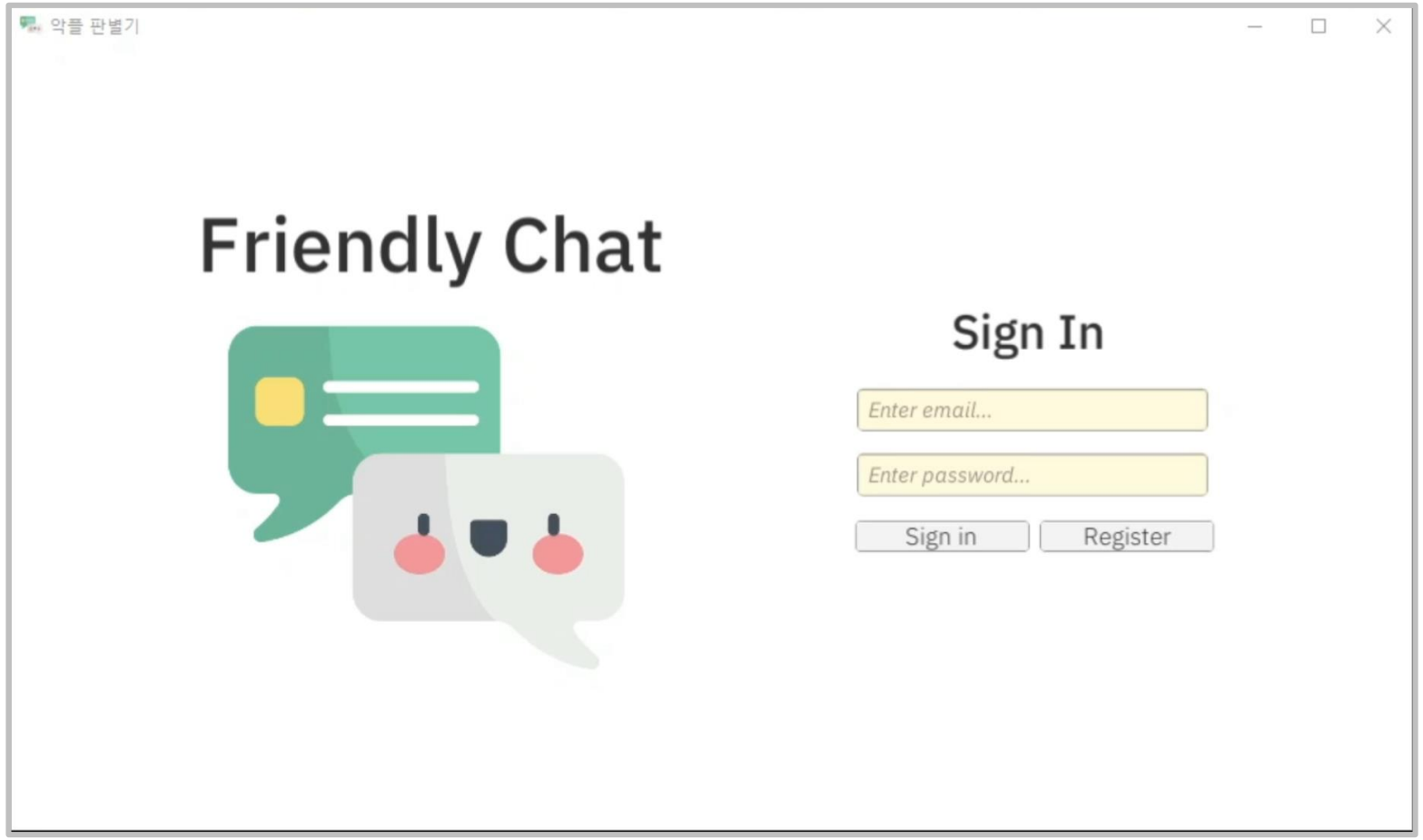
Application Logs

```
2020-01-16T02:22:26.403475+00:00 app[web.1]: 10.9.226.141 - - [16/Jan/2020:02:22:26 +0000] "GET /st
analysis.herokuapp.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_1) AppleWebKit/537.36 (KHTML
2020-01-16T02:56:37.813976+00:00 heroku[web.1]: Idling
2020-01-16T02:56:37.819679+00:00 heroku[web.1]: State changed from up to down
2020-01-16T02:56:38.557938+00:00 heroku[web.1]: Stopping all processes with SIGTERM
2020-01-16T02:56:38.569745+00:00 app[web.1]: [2020-01-16 02:56:38 +0000] [18] [INFO] Worker exiting
2020-01-16T02:56:38.569760+00:00 app[web.1]: [2020-01-16 02:56:38 +0000] [10] [INFO] Worker exiting
2020-01-16T02:56:38.569771+00:00 app[web.1]: [2020-01-16 02:56:38 +0000] [4] [INFO] Handling signal
2020-01-16T02:56:38.971100+00:00 app[web.1]: [2020-01-16 02:56:38 +0000] [4] [INFO] Shutting down:
2020-01-16T02:56:39.067688+00:00 heroku[web.1]: Process exited with status 0
2020-01-16T06:17:18.000000+00:00 app[api]: Build started by user dury.ko@gmail.com
2020-01-16T06:18:29.346393+00:00 app[api]: Deploy 786e9a20 by user dury.ko@gmail.com
2020-01-16T06:18:29.346393+00:00 app[api]: Release v38 created by user dury.ko@gmail.com
2020-01-16T06:18:30.200251+00:00 heroku[web.1]: State changed from down to starting
2020-01-16T06:18:38.825291+00:00 heroku[web.1]: Starting process with command `gunicorn app:app`
2020-01-16T06:18:40.867451+00:00 app[web.1]: [2020-01-16 06:18:40 +0000] [4] [INFO] Starting gunico
2020-01-16T06:18:40.872531+00:00 app[web.1]: [2020-01-16 06:18:40 +0000] [4] [INFO] Listening at: h
2020-01-16T06:18:40.872539+00:00 app[web.1]: [2020-01-16 06:18:40 +0000] [4] [INFO] Using worker: s
2020-01-16T06:18:40.873317+00:00 app[web.1]: [2020-01-16 06:18:40 +0000] [10] [INFO] Booting worker
2020-01-16T06:18:40.917041+00:00 app[web.1]: [2020-01-16 06:18:40 +0000] [18] [INFO] Booting worker
```

https://comment-analysis.herokuapp.com/

프론트 개발

Unity를 이용하여 프론트 개발

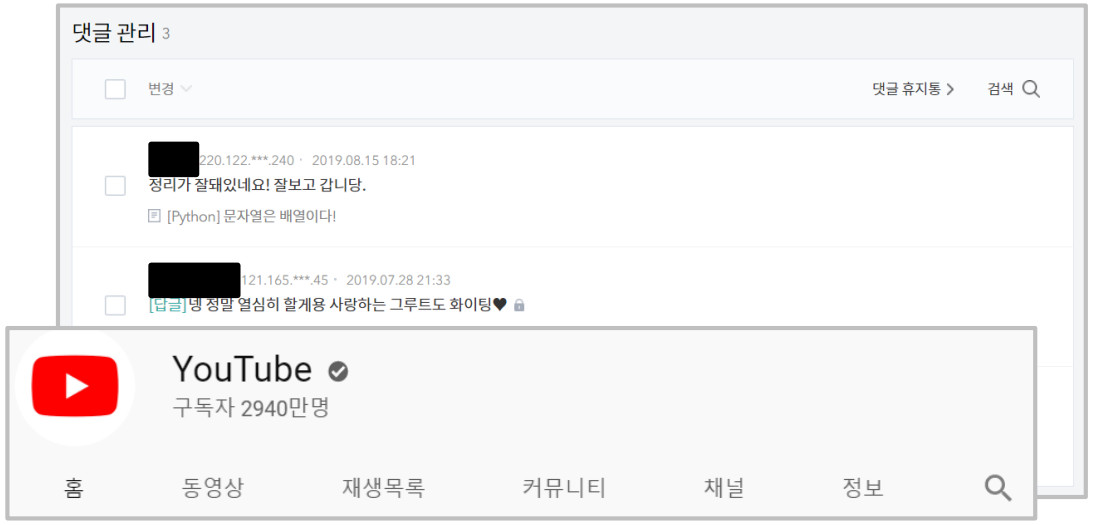


댓글을 다는 사람 입장

여러 SNS 사용 시 불쾌한 댓글을
사용자 설정에 따라서 필터링

댓글이 달리는 사람의 입장

자신에게 상처를 주는 댓글을
사용자 설정에 따라 필터링



자연어 처리가 처음이라 문자열 데이터를 어떻게 다뤄야 할지 감이 잡히지 않았다

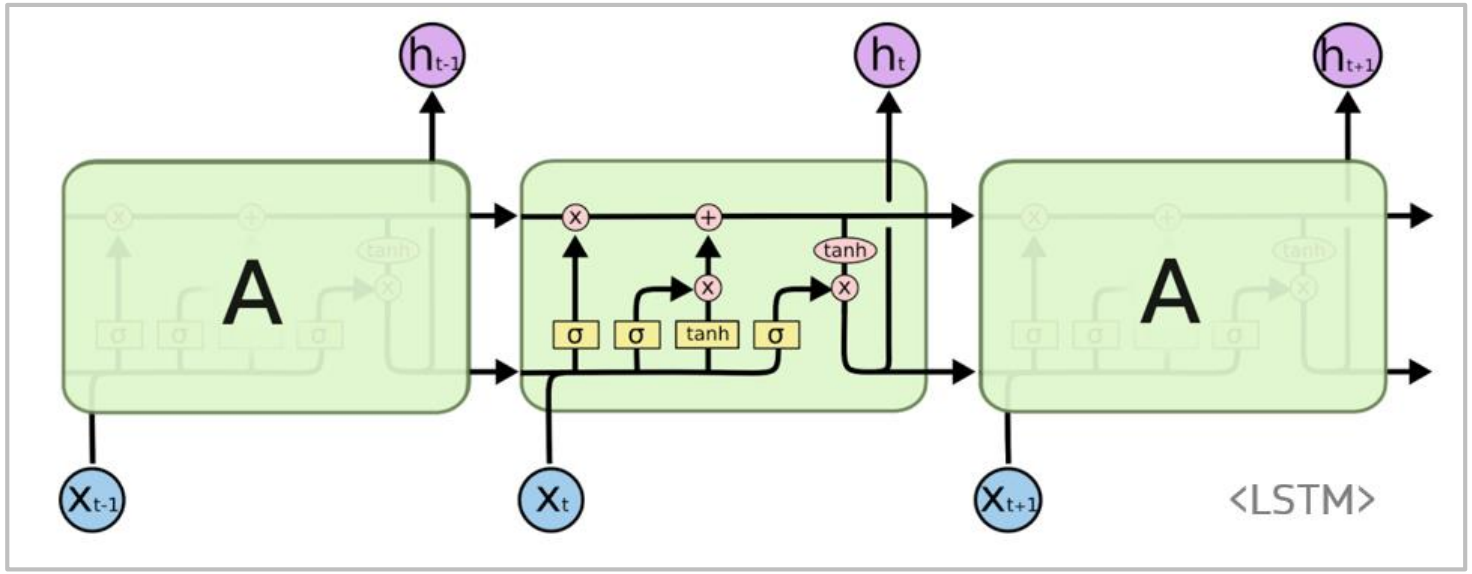
단어 개수, 문장 개수, 불용어 개수, Punctuation 개수 등 여러 피처를 생성했는데 정확도가 떨어져 코드를 다 갈아 엎었다

서버 개발을 할 때 머신러닝을 위한 서버 개발이 어려웠다 처음이라 계속해서 Error가 발생했다

04

개선할 점

문맥 파악이 전혀 되지 않아 비속어로 욕이 쓰인 경우 악성 댓글로 잡히고
욕은 쓰지 않았지만 비꼬는 댓글은 잡히지 않는다



https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/

딥러닝을 통한 문맥 파악으로성능 개선이 필요

THANK YOU