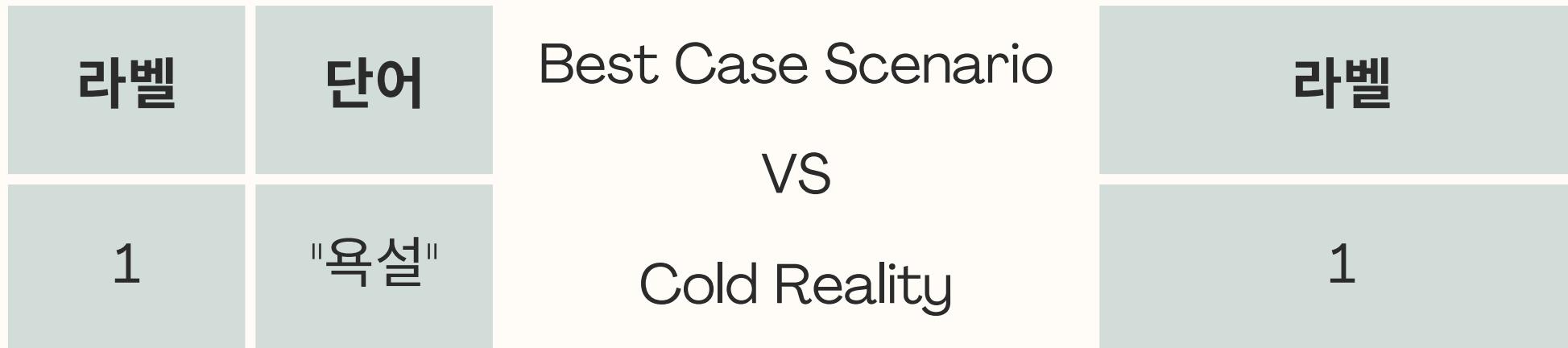


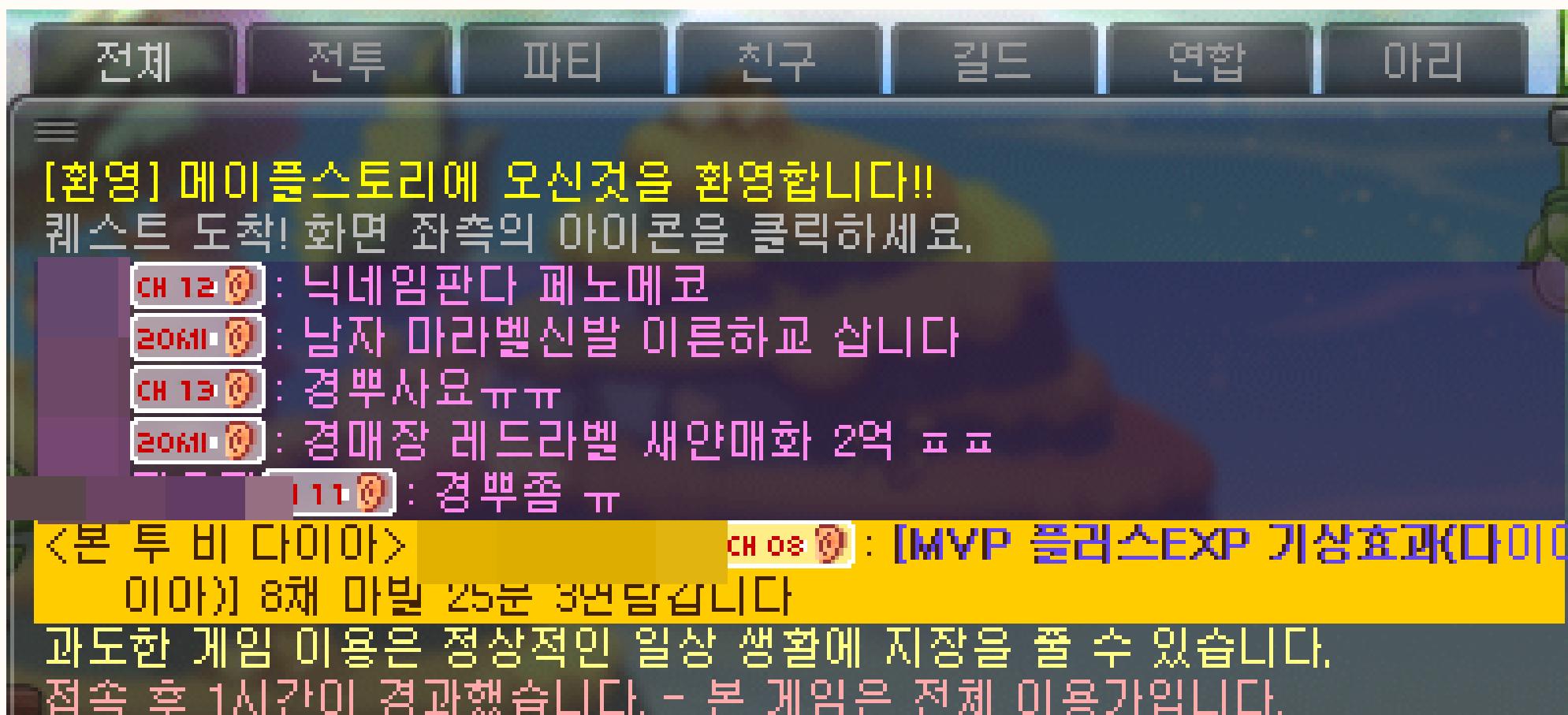
# 컨텍스트 기반 옥설 필터링

곽민규 김현호 이세진 이성현 정회수

"이 문장에 욕설이 포함되어 있습니다"



문제: 실제 기업에서도 실시간 필터링이 아닌 지속적인 모니터링을 통해 DB에 있는 모델을 학습 후 업데이트



## 프로젝트 방향성

주어진 데이터를 최대한 활용하자!

대안: 게임 인벤 크롤링 & 모델 적용

번호	제목	글쓴이
2294355	[수다] 유니온 10500 [72]	31 이노시스향기
2294290	[수다] 아기 해달 모자 귀여움 [46]	6 라비토
2294247	[수다] 우리나라가 활의 민족인 이유.EU [57]	5 메이플팁
2294212	[수다] 대리 문제 개심각한데 왜 불 안타는지 모르겠음 [139]	38 어서오렴
2294120	[수다] 메이플 부루마블 출연진.jpg [111]	72 천상에반
2293952	[수다] 할머니 리어카 끌어주는 나로 만화.manhwa [57]	52 미쳤냥
2293926	[수다] ○ㅂ뜨뜨 돌깎는거 개웃기네 ㅋㅋ [49]	70 Wraith
2293913	[수다] 서울과학고 입학한 초등학생 영재 학폭 가해자 신...	7 Xionell
2293877	[수다] 마감)현타 온 기념 이니이벤트 [84]	57 운영
2293796	[수다] 메이플 비상!!!!!! [101]	80 여행길
2293717	[수다] 이덴티스크 주민들한테 암살당할뻔했다 [23]	58 Digest
2293695	[수다] 진심으로 한입먹고 갖다 버렸다 [63]	45 히메지

# 필터링이 왜 필요할까?

## 지식 공유와 교육

건전한 환경은 정보와 유익한 지식 공유를 활성화하여 학습의 기회 제공



## 사회적 책임

자라나는 아이들에게 긍정적인 성장 모델을 제시



## 불필요한 갈등 예방

상호존중을 바탕으로 안전하고 긍정적인 커뮤니티 구축에 필수



## 다양성과 포용성 강화

다양한 배경과 견해를 가진 사람들을 위한 교류의 장 형성



# 어떤 부분에 집중했나?



Hard Positive

실제로 욕이지만  
욕이 아닌 것 같은 문장



Hard Negative

평범한 말이지만  
욕 같은 문장

# WORKFLOW

Topics Covered

1

Dataset

2

Model

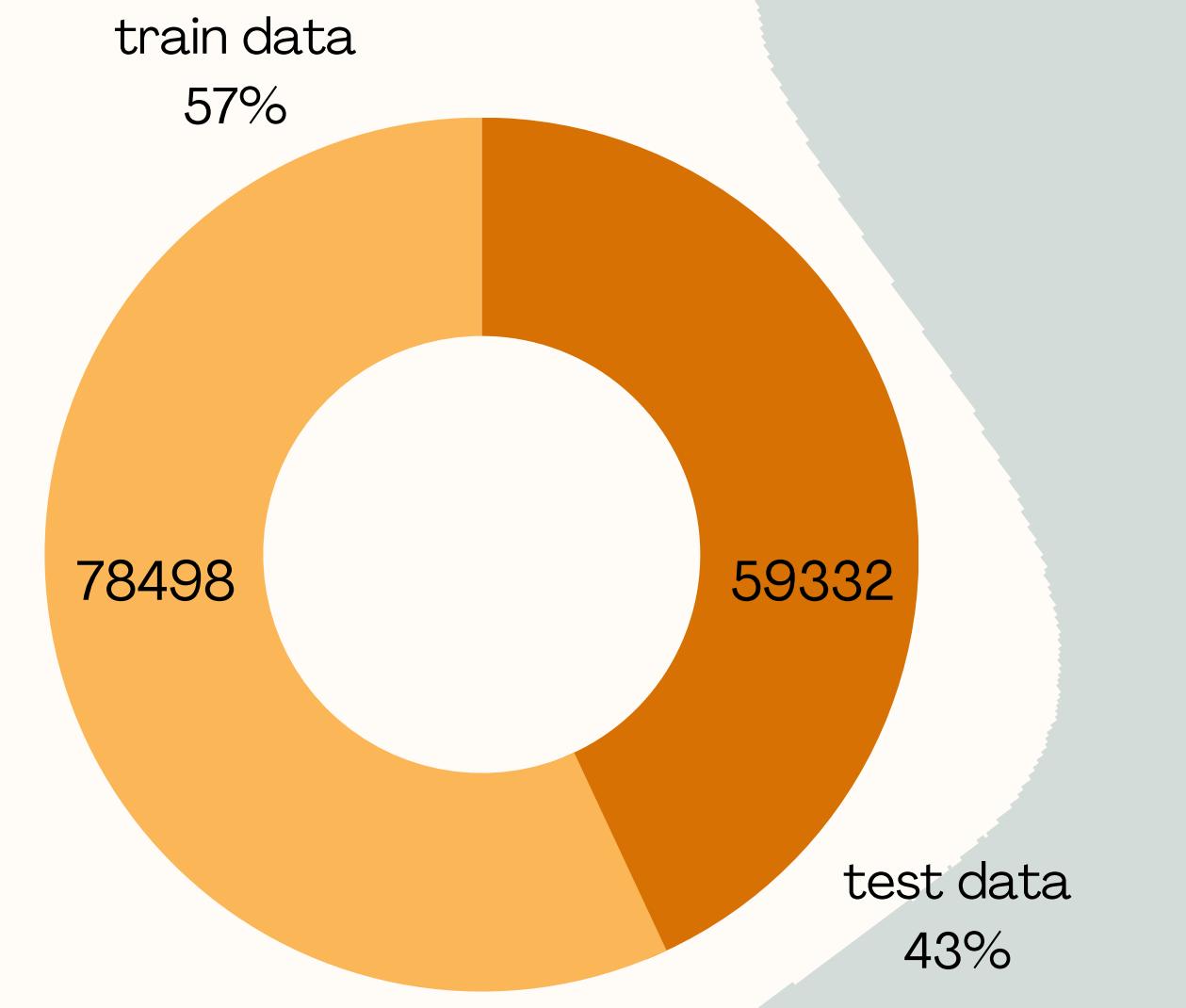
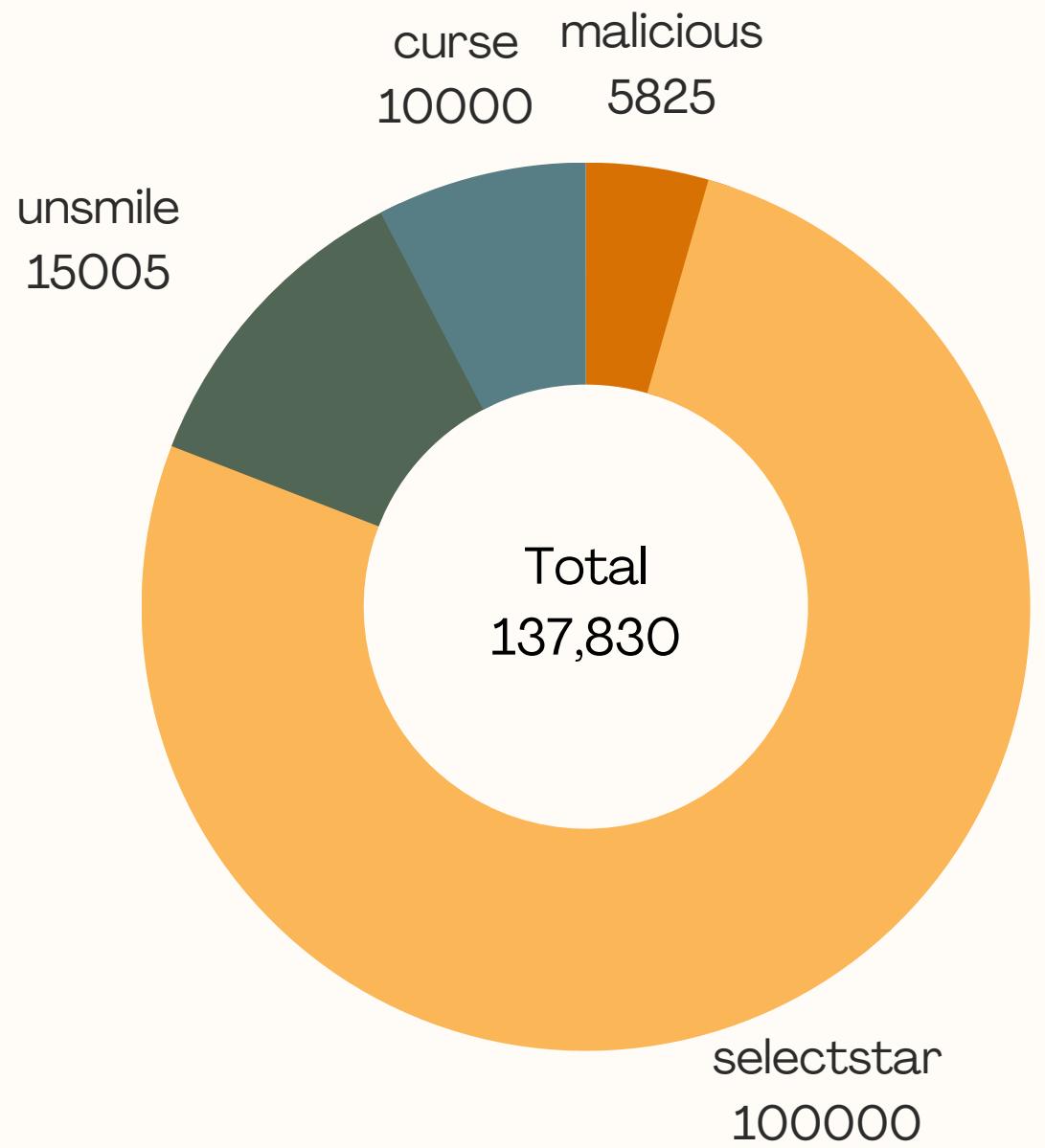
3

Crawling

4

Front

# Dataset



train data  
욕설 비율

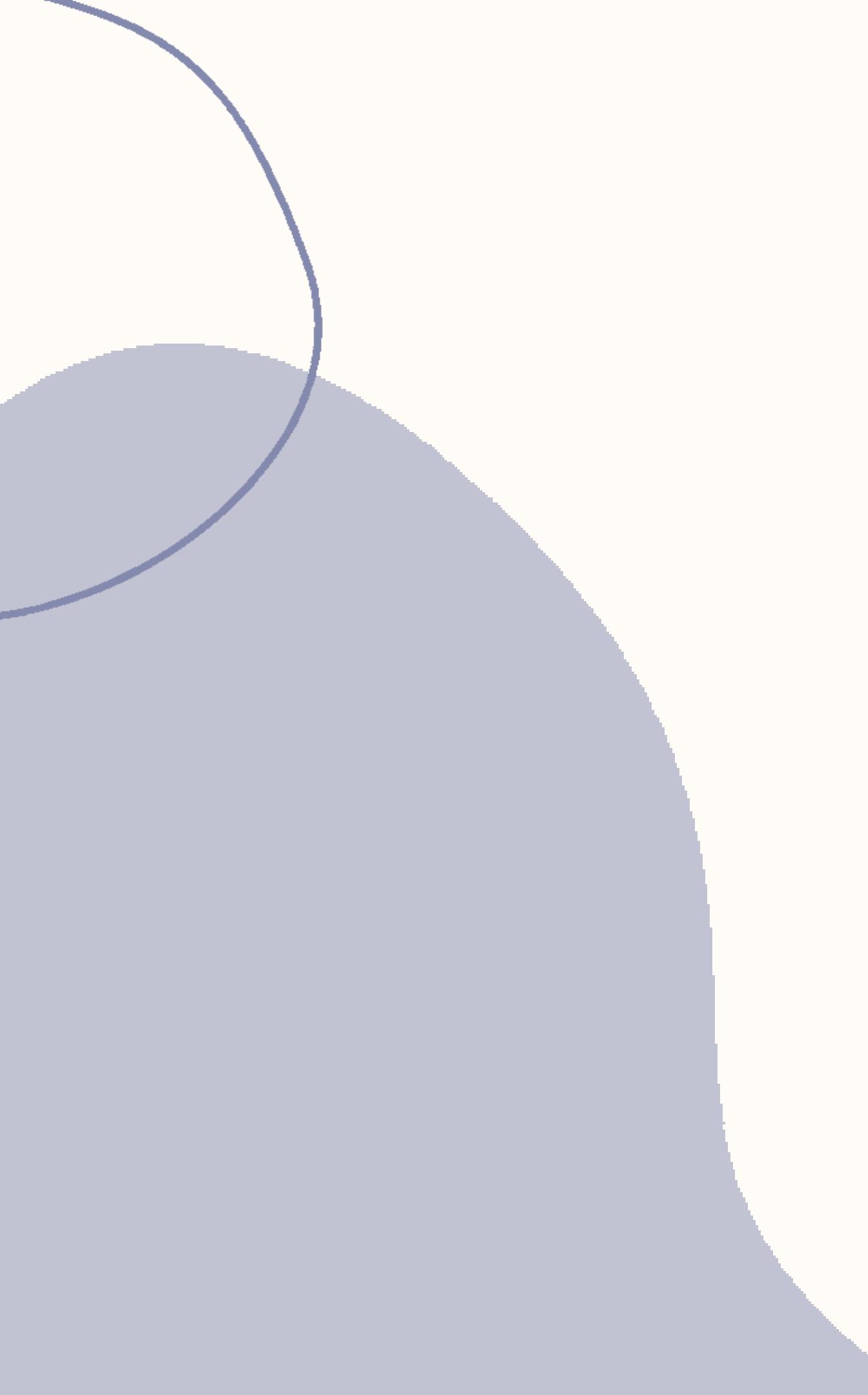
62.9%

test data  
욕설 비율

67.2%

# 통합 테이터셋 예시

	content	label
50	나이 헛 쳐드신거지 인생무상 모르니 돈을 지고 갈것도 아니고 죽으면 그만인데 자손 ...	1
51	국민의 짐아 끝까지 포기하지 말고 입당시켜라 ㅋㅋ 개털이라고 헌신짝처럼 버리지말고...	0
52	저런 쓰레기같은 인간들.제발 한국땅에 들어오지못하게해라.	1
53	정부였으면 폭동 났다 광화문에 .. 등등 연예인 총출동 시켜서 촛불잔치 하고도 남...	0
54	민노총은 사악한집단.없어져야한다.	0
55	미구는 가능할걸	0
56	게네들 영구 제명시키고 이군 선수들 콜업해라 ? 노름,마약,술판,성희롱등은 많은 돈...	0
57	영처리 상황파악 안되네...마 지금 코로나와 경제폭망, 치솟는 실업률로 남한 사람들...	1
58	설마 했는데 역시 중공이었네... 짱깨들 맥주40병 쳐마신게 ...왜 ...	1



# Model

# 선정 모델 및 최적화 과정

## SKTBrain/KoBERT

Korean BERT pre-trained cased (KoBERT)



- 한국어 뉴스기사와 커뮤니티 댓글 기반 pre-trained model.
- 감정 모델의 아키텍처를 차용하여 욕설을 분류하는 함수 추가

### 1. 모델 개선: 원본 모델에 Dense Layer를 한 층 더 추가하여 언어 처리의 성능을 개선

```
opt = tfa.optimizers.RectifiedAdam(lr=0.00005, total_steps = 5600, warmup_proportion=0.08, min_lr=0.000001, epsilon=0.01, clipnorm=0.9)

sentiment_drop = tf.keras.layers.Dropout(0.3)(bert_outputs)
dense_layer = tf.keras.layers.Dense(64, activation='relu')(sentiment_drop)
sentiment_drop2 = tf.keras.layers.Dropout(0.3)(dense_layer)
dense_layer2 = tf.keras.layers.Dense(8, activation='relu')(sentiment_drop2)
sentiment_drop3 = tf.keras.layers.Dropout(0.3)(dense_layer2)
sentiment_first = tf.keras.layers.Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.TruncatedNormal(stddev=0.02))(sentiment_drop3)
sentiment_model = tf.keras.Model([token_inputs, mask_inputs, segment_inputs], sentiment_first)
sentiment_model.compile(optimizer=opt, loss=tf.keras.losses.BinaryCrossentropy(), metrics = ['accuracy'])
```

## 2. Hyperparameter 조절 과정:

- Validation accuracy가 0.6290에서 개선되지 않는 상황이 자주 발생했고 training set과 일치하는 유탈 비율을 보인 것을 바탕으로 학습이 안 되고 있다고 판단.

```
sentiment_model.fit(train_x, train_y, epochs=8, shuffle=True, batch_size=64, validation_data=(test_x, test_y))

Epoch 1/8
1227/1227 [=====] - 1404s 1s/step - loss: 0.6684 - accuracy: 0.6248 - val_loss: 0.6595 - val_accuracy: 0.6294
Epoch 2/8
1227/1227 [=====] - 1331s 1s/step - loss: 0.6599 - accuracy: 0.6281 - val_loss: 0.6533 - val_accuracy: 0.6290
Epoch 3/8
1227/1227 [=====] - 1335s 1s/step - loss: 0.6533 - accuracy: 0.6356 - val_loss: 0.6439 - val_accuracy: 0.6530
Epoch 4/8
1227/1227 [=====] - 1335s 1s/step - loss: 0.6588 - accuracy: 0.6339 - val_loss: 0.6599 - val_accuracy: 0.6290
Epoch 5/8
1227/1227 [=====] - 1329s 1s/step - loss: 0.6628 - accuracy: 0.6289 - val_loss: 0.6596 - val_accuracy: 0.6290
Epoch 6/8
1227/1227 [=====] - 1335s 1s/step - loss: 0.6626 - accuracy: 0.6290 - val_loss: 0.6598 - val_accuracy: 0.6290
Epoch 7/8
1227/1227 [=====] - 1335s 1s/step - loss: 0.6626 - accuracy: 0.6290 - val_loss: 0.6595 - val_accuracy: 0.6290
Epoch 8/8
1227/1227 [=====] - 1335s 1s/step - loss: 0.6623 - accuracy: 0.6290 - val_loss: 0.6595 - val_accuracy: 0.6290
<keras.callbacks.History at 0x7a4813d0b3a0>
```

```
from sklearn.metrics import classification_report
y_true = test['label']
# F1 Score 확인
print(classification_report(y_true, np.round(preds,0)))

precision    recall  f1-score   support
          0       0.68      1.00      0.77     49373
          1       0.00      0.00      0.00     29125
accuracy                           0.63     78498
macro avg       0.31      0.50      0.39     78498
weighted avg    0.40      0.63      0.49     78498
```

- Learning Rate를 키우고, 초기에 높은 학습률을 보이게 하는 warmup\_proportion을 0.15로 설정
- 노드 학습 성능 개선을 위해서 dropout을 0.05 낮춰 노드가 학습하는 비율을 높임(0.30 → 0.25).
- minimum learning rate을 learning rate의 2% 수준으로 낮춰 성능 향상

```
# 총 batch size * 4 epoch = 2344 * 4
opt = tfa.optimizers.RectifiedAdam(lr=0.00005, total_steps = 6000, warmup_proportion=0.15, min_lr=0.000001, epsilon=1e-08, clipnorm=0.9)
```

```
sentiment_drop = tf.keras.layers.Dropout(0.25)(bert_outputs)
dense_layer = tf.keras.layers.Dense(64, activation='relu')(sentiment_drop)
sentiment_drop2 = tf.keras.layers.Dropout(0.25)(dense_layer)
dense_layer2 = tf.keras.layers.Dense(8, activation='relu')(sentiment_drop2)
sentiment_drop3 = tf.keras.layers.Dropout(0.25)(dense_layer2)
sentiment_first = tf.keras.layers.Dense(1, activation='sigmoid', kernel_initializer=tf.keras.initializers.TruncatedNormal(stddev=0.02))(sentiment_drop3)
sentiment_model = tf.keras.Model([token_inputs, mask_inputs, segment_inputs], sentiment_first)
sentiment_model.compile(optimizer=opt, loss=tf.keras.losses.BinaryCrossentropy(), metrics = ['accuracy'])
```

### 3. Optimization 과정 :

- minimum learning rate, steps, warmup proportion, epsilon, clipnorm, activation function, optimizer, epoch)를 조절하며 모델 성능을 관찰
- relu, tanh, leaky relu 등 여러 개의 Activation function 시도
- 적절한 조합을 찾은 뒤에는 epoch, steps, random seed를 설정하며 최고의 성능을 보이는 지점을 관찰

moderate

moderate

best

No	작성자	accuracy	F1 macro avg	learning_rate	activate_func	optimizer	epoch	epsilon	clipnorm	steps	hup_propo	min_lr	loss	threshold	dropout	random_seed	
1	이성현	0.7957		0.001	relu	RectifiedAdam	4	0.008	1	0.00001	8000	0.1	0.471	0.94			
2	이세진	0.63	0.39	0.001	relu	Adam	6	1.00E-05	1				0.6595	상관없이 성능X			
3	이세진	0.62	0.63	5.00E-05	relu	Adam	6	1.00E-08	1				0.66	0.94			
4	이세진	0.62	0.63	5.00E-05	relu	Adam	6	1.00E-08	1				0.66	0.94			
5	이성현	0.8225	0.85	0.0025	relu	RectifiedAdam	5	1.00E-02	1	0.00001	10000	0.1	0.4214	0.94			
6	이세진	X	X	5.00E-05	relu	RectifiedAdam	10	1.00E-08	1	2344*2	0.1	1.00E-05	X	X			
7	이세진	0.79	0.81	5.00E-05	leakyrelu	RectifiedAdam	10	1.00E-08	1	2344*2	0.1	1.00E-05	0.6457	0.94			
8	곽민규			0.001	tanh	RectifiedAdam	10	1.00E-08	1	2344*2	0.1	0.00001					
9	이성현	0.8763	0.91	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.3168	0.94	0.25		
10	이세진	0.62		전부 기본									0.6595	0.3206059337			
11	이세진	0.86	0.84	전부 기본									0.6595	0.3206059337			
12	이성현	0.7549	0.74	0.00005	relu	RectifiedAdam	9	1.00E-02	0.9	5600	0.08	0.000001	0.5386	0.94			
13	이성현	0.629	0.39	0.000075	relu	RectifiedAdam	7	1.00E-02	0.9	7500	0.1	0.000001	0.6434	0.94			
14	이성현	0.629	0.39	0.00005	relu	RectifiedAdam	10	1.00E-02	0.9	6000	0.08	0.000001	0.6617	0.94			
15	이성현	0.71	0.7417	0.00008	leaky_relu	RectifiedAdam	8	1.50E-02	0.85	7000	0.08	0.000025	0.5596	0.94			
16	이성현	0.8309	0.84	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.4288	0.94			
17	이성현	0.629	0.39	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6621	0.94			
18	이성현	0.629	-	0.00075	relu	RectifiedAdam	5	1.00E-08	0.9	7000	0.08	0.000001	0.4212	0.94	0.25	6	ting ra
19	이성현	0.629	0.39	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6621	0.94	0.25	1	
20	이성현	0.629	0.39	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6621	0.94	0.25	2	
21	이성현	0.629	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	-	0.94	0.25	3	3 epoch
22	이성현	0.629	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	-	0.94	0.25	4	3 epoch
23	이성현	0.6931	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.66	0.94	0.25	5	단정지
24	이성현	0.8379	0.83	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.4212	0.94	0.25	6	에서는
25	이성현	0.6931	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6622	0.94	0.25	7	3 epoch
26	이성현	0.6931	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6622	0.94	0.25	8	3 epoch
27	이성현	0.6931	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6622	0.94	0.25	9	3 epoch
28	이성현	0.6934	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	-	0.94	0.25	10	3 epoch
29	이성현	0.629	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6635	0.94	0.25	11	3 epoch
30	이성현	0.6995	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6635	0.94	0.25	12	5 epoch
31	이성현	0.7039	0.62	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6067	0.94	0.25	13	
32	이성현	0.629	-	0.00005	relu	RectifiedAdam	5	1.00E-08	0.9	6000	0.08	0.000001	0.6624	0.94	0.25	14	4 epoch
33	곽민규	0.707	0.71	0.00005	relu	RectifiedAdam	8	1.00E-08	0.9	6000	0.08	0.000001	0.6046	0.94	0.25	16	목설 0
34	곽민규	-	-													17	
35	곽민규	0.629	-	0.00005	relu	RectifiedAdam	8	1.00E-08	0.9	6000	0.08	0.000001	0.6599	0.94	0.25	18	3 epoch
36	곽민규	-	-													19	
37	곽민규	0.629	-	0.00005	relu	RectifiedAdam	8	1.00E-08	0.9	6000	0.08	0.000001	0.6595	0.94	0.25	20	3 epoch
38	곽민규	-	-													21	
39	곽민규	-	-													22	
40	곽민규	0.629	0.63	0.00005	relu	RectifiedAdam	8	1.00E-08	0.9	6000	0.08	0.000001	0.6437	0.94	0.25	23	
41	곽민규	-	-													24	
42	곽민규	-	-													25	
43	이세진	X	사용불가													58	
44	이세진	X	사용불가													53	
45	이세진	X	사용불가													32	
46	이세진	X	사용불가													28	
47	이세진	0.889	0.88	0.00005	relu	RectifiedAdam	8	1.00E-08	0.9	6000	0.08	0.000001	0.2865	0.45	0.25	13	조정해!
48	이세진	0.8879	0.91	0.00005	relu	RectifiedAdam	10	1.00E-08	0.9	6000	0.08	0.000001	0.2874	0.55	0.25	13	조정해!
49	이성현			0.00005	relu	RectifiedAdam	12	1.00E-08	0.9	6000	0.08	0.000001	0.2874	0.94	0.25	13	컬 변
50	이성현	0.9333	96	0.00005	relu	RectifiedAdam	20	1.00E-08	0.9	6000	0.08	0.000001	0.1903	0.94	0.25	13	최

# 초기 모델 VS. 최종 모델

- Overfitting된 기존 모델 (epoch만 조절 4 → 10)

```
[72] sentiment_model.fit(train_x, train_y, epochs=10, shuffle=True, batch_size=64, validation_data=(test_x, test_y))  
45  
Epoch 10/10  
235/235 [=====] - 288s 1s/step - loss: 0.0458 - accuracy: 0.9897 - val_loss: 1.0845 - val_accuracy: 0.7728  
<keras.callbacks.History at 0x7c59a84da6b0>
```

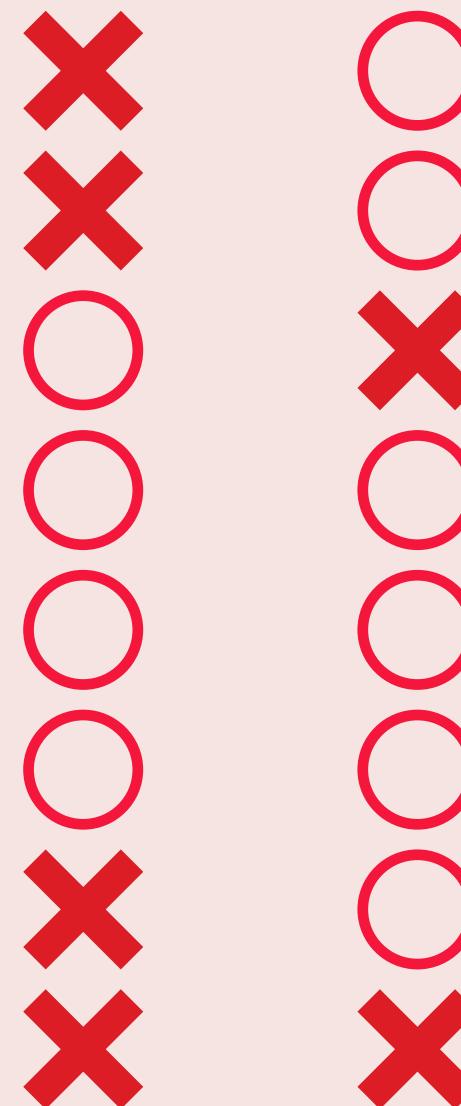
- 최종 구현된 모델 성능(overfitting 없이 validation accuracy가 96.35%로 나타남)

```
] sentiment_model.fit(train_x, train_y, epochs=20, shuffle=True, batch_size=64, validation_data=(test_x, test_y))  
Epoch 20/20  
1227/1227 [=====] - 1327s 1s/step - loss: 0.1903 - accuracy: 0.9333 - val_loss: 0.1147 - val_accuracy: 0.9635  
<keras.callbacks.History at 0x7d4310f63d00>
```

## 초기 모델 실제 사례 적용(50% 정확)

```
evaluate_text("겜 그따구로 하지좀마")
evaluate_text("느금마")
evaluate_text('띠발')
evaluate_text('씨발')
evaluate_text('씨2발')
evaluate_text('개같은새끼야')
evaluate_text('전염병')
evaluate_text('자지마')
```

```
1/1 [=====] - 0s 73ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 76ms/step
(욕설 확률 : 0.68) 입력 문장은 욕설이 아닐 가능성이 높습니다.
1/1 [=====] - 0s 79ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 80ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 84ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 169ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 104ms/step
(욕설 확률 : 0.98) 입력 문장은 욕설일 가능성이 있습니다.
1/1 [=====] - 0s 147ms/step
(욕설 확률 : 1.00) 입력 문장은 욕설일 가능성이 있습니다.
```



## 최적화 모델 실제 사례 적용(75% 정확도)

```
evaluate_text("겜 그따구로 하지좀마")
evaluate_text("느금마")
evaluate_text('띠발')
evaluate_text('씨발')
evaluate_text('씨2발')
evaluate_text('개같은새끼야')
evaluate_text('전염병')
evaluate_text(['자지마'])
```

```
1/1 [=====] - 0s 134ms/step
(욕설 확률 : 0.02, 욕설 아닐 확률 : 0.10) 욕설 XXX
1/1 [=====] - 0s 130ms/step
(욕설 확률 : 0.99) 욕설 000
1/1 [=====] - 0s 185ms/step
(욕설 확률 : 0.03, 욕설 아닐 확률 : 0.10) 욕설 XXX
1/1 [=====] - 0s 98ms/step
(욕설 확률 : 0.99) 욕설 000
1/1 [=====] - 0s 86ms/step
(욕설 확률 : 0.99) 욕설 000
1/1 [=====] - 0s 74ms/step
(욕설 확률 : 0.99) 욕설 000
1/1 [=====] - 0s 71ms/step
(욕설 확률 : 0.01, 욕설 아닐 확률 : 0.10) 욕설 XXX
1/1 [=====] - 0s 76ms/step
(욕설 확률 : 0.99) 욕설 000
```

# **욕설 주의!**

모델 성능 임의 검증을 위해 각종 욕설 문구를 대상으로 실험한 내용을 공유할 예정입니다.

프로젝트의 원활한 진행을 위한 과정이니 너른 마음으로 양해 부탁드립니다.

# 모델 성능 임의 검증

```
hard_positive = ['개샛기', '개색기', 'c8색기', '씨foot 네가 뭐라도 되는 줄 아나', '시foot 재밌네ㅋㅋㅋ', '시8',  
    'bottle god', 'sea pearl', 'G랄', 'Z랄', '빙신', '별진', '시 발', '식발', '씨이발', '씨이발', '찐',  
    '₩ls', '^~|발', '^|발', '^1발', '^~1발', '좆망겜', '개좆망겜', '률신', '승가', '스바', '젖통',  
    '우유통', '빨통', '띠발', 'Tlakf', 'tlakf', '느금마', '재기해', '씨2발']  
soft_positive = ['지랄', '병신', '찐따', '시발', '썅', '엠병', '꺼져', '씹', '고자', '좆망겜', '보자',  
    '자자', '개년', '자위', '딜도', '말가', '성기', '자궁', '브라', '예로', '음부', '가슴']  
soft_negative = ['나라사랑카드', '삼다수 물병', '달력이 이제 몇월을 가리키고 있지?', '지갑을 잊어버렸다',  
    '캐리어를 들고 공학관 D503호로 오세요', '최종발표는 화요일에 진행됩니다']  
hard_negative = ['처음부터', '사슴가죽', '빨리 해결해야 한다', '피예로', '이거 씹어먹지마', '염구 없다',  
    '게임하고자 해', '한조까지 있네', '이거 어딨는지 뒤져 봐', '앱이 자꾸 꺼져', '이젠 장난 그만해',  
    '함재기', '시발점', '경제개발5개년 계획', 'ㅄ발침', '풀스바겐 걸로', '화염병', '아구예로',  
    '너 남자지?', '백년묵은 버섯', '어딜도망가', '살자위', '전성기', '수호자궁으로 잡으세요', '브라질']
```

### 잘못 예측한 예시

'띠발', '자지마', '보지마', '자1지마', '보1지마', 'tlqkf', 'Tlqkf'

### ##### hard positive #####

'씨foot 네가 뭐라도 되는 줄 아나' '시foot 재밌네ㅋㅋㅋ', 'bottle god', 'sea pearl', 'Wls', '^\_^ | 발', '좆망겜', '슴가', '스바', '젖통', '우유통', '빨통', '띠발', 'Tlqkf', 'tlqkf', '재기해'

### ##### soft positive #####

'쌍', '고자', '자위', '딜도', '성기', '브라', '에로', '음부', '가슴'

### ##### soft negative #####

<해당사항 없음.>

### ##### hard negative #####

'시발점', '상자위'

### <잘 예측한 예시>

"겜 그따구로 하지좀마", "느금마", '씨발', '씨2발', '개같은새끼야', '전염병'

### ##### hard positive #####

개샛기', '개색기', 'c8색기', '시8', 'G랄', 'Z랄', '빙신', '병진', '시 발', '식발', '씨이발', '씨@발', '찐', '^\_^ | 발', 'ㅅ1발', '^\_^1발', '개좆망겜', '붕신', '느금마', '씨2발'

### ##### soft positive #####

지랄', '병신', '찐따', '시발', '엠병', '꺼져', '씹', '좆망겜', '보지', '자자', '개년', '망가', '자궁',

### ##### soft negative #####

'나라사랑카드', '삼다수 물병', '달력이 이제 몇월을 가리키고 있지?', '지갑을 잃어버렸다', '캐리어를 들고 공학관 D503호로 오세요', '최종발표는 화요일에 진행됩니다'

### ##### hard negative #####

'처음부터', '사슴가죽', '빨리 해결해야 한다', '피에로', '이거 씹어먹지마', '영구 없다', '게임하고자 해', '한조까지 있네', '이거 어딨는지 뒤져 봐', '앱이 자꾸 꺼져', '이젠 장난 그만해', '함재기', '경제개발5개년 계획', '뼈받침', '폭스바겐 걸로', '화염병', '아구에로', '너 남자지?', '백년묵은 버섯', '어딜도망가', '전성기', '수호자궁으로 잡으세요', '브라질'

hard positive data : 35

soft positive data : 23

soft negative data : 6

hard negative data : 25

정확도	soft	hard
positive	13/23 (56.52%)	20/35 (57.14%)
negative	6/6 (100%)	23/25 (92%)

# Crawling

번호	제목
2296205	[수다] 편의성 개선좀 합시다!!! [52]
2296183	[수다] 분탕들 존나 귀여워지긴 했네 [41]
2296170	[인방] ○ ㅂ)춘자 5.5메르 검술 [128]
2296099	[수다] 에테르넬 24성 도적모자팝니다 [108]
2295853	[수다] 유라라 [47]

```
## 정적 크롤링으로 화제글 위에서부터 10개 뽑아오기
html_table = requests.get(url).text
soup_table = bs(html_table, 'html.parser')
data_box = soup_table.find_all('a', attrs = {'class' : 'subject-link'})[:10]
```

- 인벤 사이트 화제글 리스트 정적 크롤링으로 추출
- 상위 10개 게시글에서 게시글 당 랜덤으로 댓글 3

개 동적 크롤링으로 추출

- 추출한 댓글들이 특정 게시글에만 의존하는 현상

을 해소하기 위해 무작위 선별 방식 채택.

53 Moky (2023-08-21 20:26:57)	34  0  공감 확인  신고
마지막꺼 진짜 좋은데	
↳ 70 루세드 작성자 (2023-08-21 20:28:39)	0  0  공감 확인  신고
ㄹㅇ 달나 다크니스오라 소마 이클립스 이런 막타 쳐야 되는 스킬 노안이슈or 명매리다 날려먹을 때마다 너무 간절해짐 ㅜ	
↳ 53 Moky (2023-08-21 20:31:56)	1  0  공감 확인  신고
나도 부캐 인피니티 자꾸 까먹음 ..	
↳ 50 루시352 (2023-08-21 20:40:31)	1  0  공감 확인  신고
오더갯수, 지속시간봐야하는데 ㄹㅇ 개좋다	
↳ 16 너무긴 (2023-08-21 20:40:44)	0  0  공감 확인  신고
그리고 스킬알림이 이거 알림 지속시간도 설정가능하면 좋겠음	
존나 빨리 사라져서 못본적 많음	
↳ 72 토건 (2023-08-21 20:42:15)	0  0  공감 확인  신고
인피 ui제발	
<pre>## Xpath 형식 동일한 댓글들 모조리 긁어오기 comment_box = driver.find_elements(     By.XPATH, '//*[starts-with(@id, "cmt")]/div[2]/div[2]/span')  ## 전체 댓글에서 3개 랜덤추출 후 결과 리스트에 추가 selected_comments = random.sample(comment_box, 3) comment_list = comment_list + [comment.text for comment in selected_comments]</pre>	

```

df_result = pd.DataFrame()
df_result[ "leagueoflegends" ] = query(0)
df_result[ "maplestory" ] = query(1)
df_result[ "fifa4" ] = query(2)
df_result[ "diablo4" ] = query(3)
df_result[ "lostark" ] = query(4)

```

리그 오브 레전드 인벤의 오늘의 화제글을 불러옵니다.

댓글 추출이 완료되었습니다!

메이플스토리 인벤의 30추글 목록을 불러옵니다.

댓글 추출이 완료되었습니다!

피파온라인 4 인벤의 오늘의 화제글을 불러옵니다.

댓글 추출이 완료되었습니다!

디아블로 4 인벤의 30추글 목록을 불러옵니다.

댓글 추출이 완료되었습니다!

로스트아크 인벤의 30추글 목록을 불러옵니다.

댓글 추출이 완료되었습니다!

leagueoflegends	maplestory	fifa4
좀 치네ㅋㅋㅋㅋㅋㅋ	그럼 사냥터 개인화에 다인사냥 터 활성화가 될 수록 매크로가 엿먹는거네?	ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ ㅋ...
자매품 : 삼대떡	미니던전 삭제하던 짬 남아있네 ㅋㅋ	홍박사님을 아세요
윗 두명다 병신인건 알겠네	거탐걸면 발작하더라	텐하호아님? ㅋㅋ
추천 올려서 많은 분들이 보게 해줍시다	19 입니다. 어려운 희망이에요. 너무 슬퍼마세요...	골반이 별론뎅
방구석에서 게임만 하는 찌끄래기 인생 이라 그저 남 욕하는 게 제일 쉬우니까 그거밖에...	오늘 좋은 운은 아니네요. 주사 위는 33 입니다...	ㄷㄷ

diablo4	lostark
설마 윙크한거??	스윗남 ㅋㅋㅋ
ㄹㅇㅋㅋ	ㅋㅋㅋㅋㅋ
재들 본 뒤론 '게임 안해본 새 키들이 디4를 논하냐' or '제대 로 키워보지도 않고...	ㅋㅋㅋㅋㅋㅋㅋㅋ ㅋㅋㅋㅋㅋㅋㅋㅋ ㅋㅋㅋㅋㅋㅋ
으 동망겜 하나하나 나열하니 까 진짜 개역겹네	바드는 메로엣타
맞는말인데 지랄 발광 떠네 병 신들 ㅋㅋㅋ	구름 포켓몬입니당...

- 문장 줄바꿈은 띄어

쓰기로 변환.

- 이미지만 있는 댓글

은 "이미지만 있는  
댓글입니다." 출력.

```
## 웹드라이버 옵션 추가: 동적 크롤링 성능 향상을 위해 쓸데없는 옵션들 모두 제외
service = Service()
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument("disable-gpu")
options.add_argument("disable-infobars")
options.add_argument("--disable-extensions")
caps = DesiredCapabilities().CHROME
caps["pageLoadStrategy"] = "none"
```

```
try:
    ## 코멘트 래퍼 로딩까지 기다리기
    WebDriverWait(driver, 10).until(EC.visibility_of_element_located((
        By.XPATH, '//*[@id="powerbbsCmt2"]/div[2]')))

except TimeoutException:
    print("요소가 나타나지 않았습니다.")
```

	룰	메이플	피파 4	디아블로 4	로스트아크
Version 2	73.79059 sec	71.51561 sec	71.72133 sec	77.46055 sec	69.96042 sec
Version 3 (single)	25.09268 sec	20.85221 sec	19.92404 sec	15.90243 sec	18.80874 sec
Version 3 (multi, 3)	32.38270 sec	27.62173 sec	28.85698 sec	26.52251 sec	36.71467 sec
Version 3 (multi, 5)	30.73513 sec	21.09598 sec	20.40007 sec	24.56444 sec	30.33779 sec
Version 3 (multi, 10)	41.86643 sec	26.64503 sec	28.09447 sec	32.85625 sec	33.20684 sec

## 소요 시간을 줄이기 위한 노력

동적 크롤링 성능 향상을 위해 Webdriver 옵션 추가

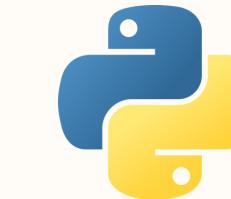
WebDriverWait 와 expected\_conditions 를 활용해서 time.sleep() 사용할 때보다 효율적인 동적 크롤링 시도 및 에러 핸들링

concurrent.futures 의 ThreadPoolExecutor를 활용해서 병렬처리 작업 시도 및 소요 시간 비교

Json 형식 파일로 변환

# Front

# BeautifulSoup



# React

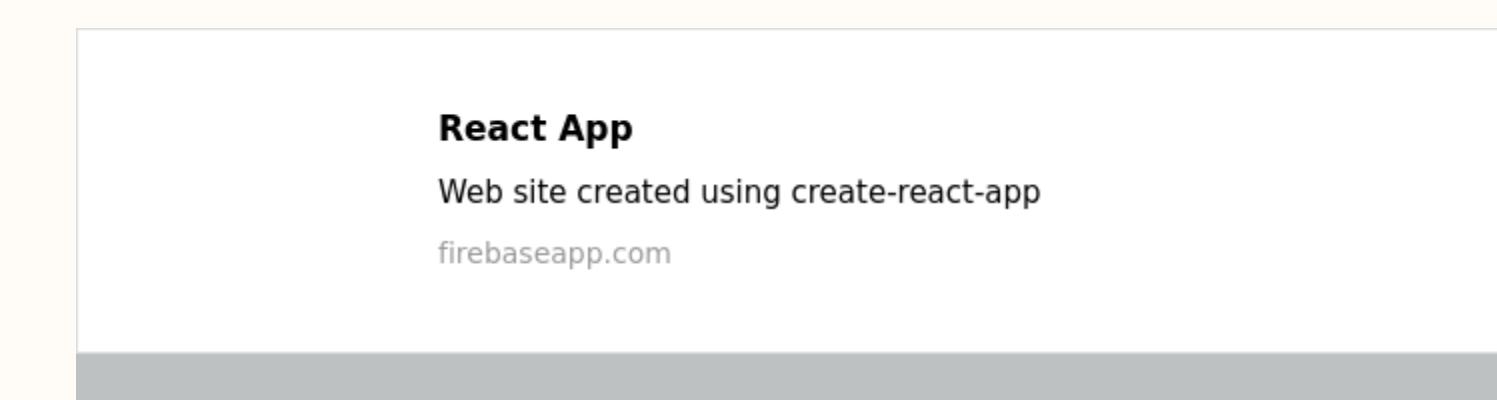
javascript 기반 React 라이브러리 +  
gcp (google cloud platform) function 기능

JSON 형식 파일



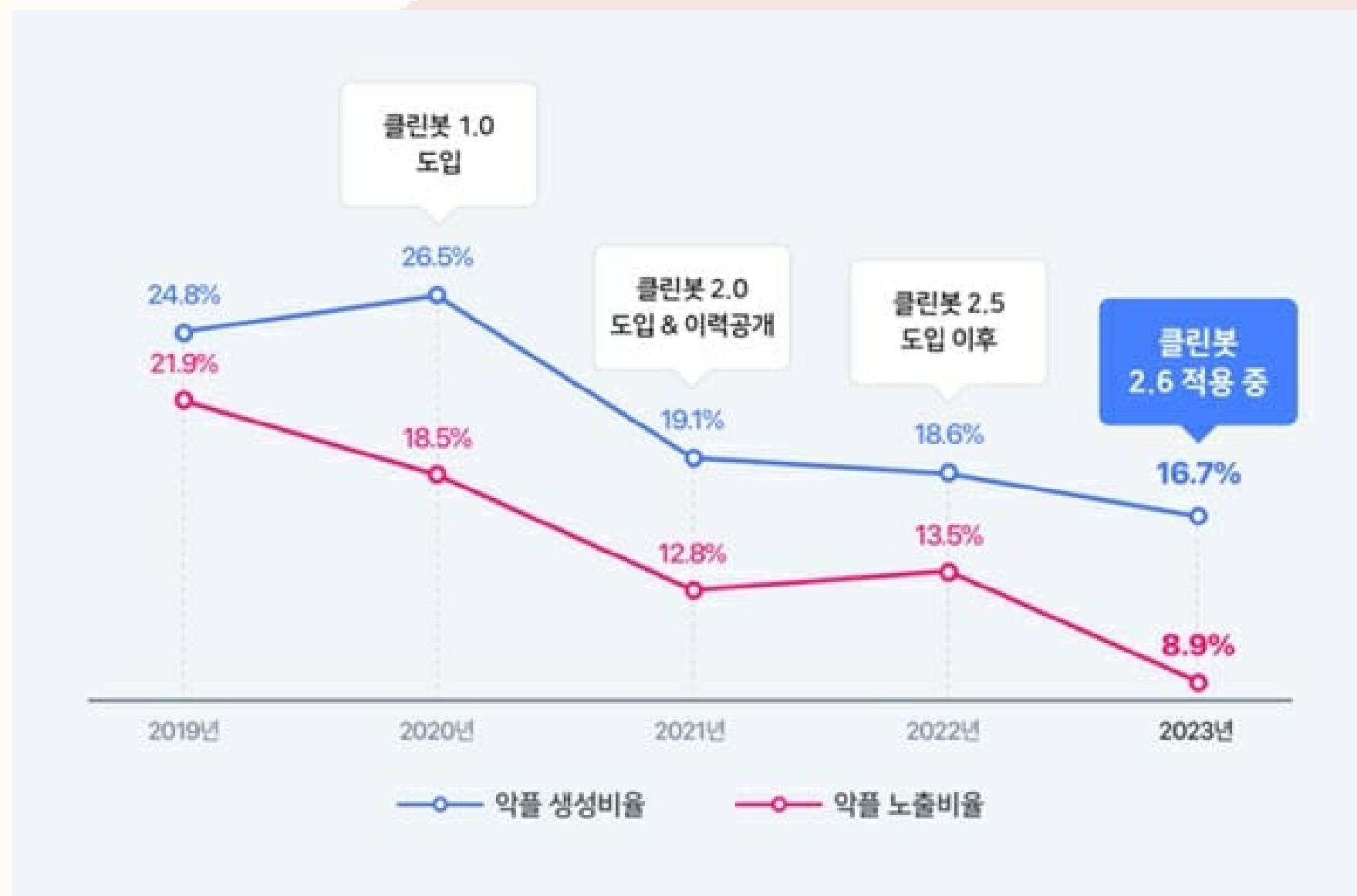
# Firebase

데이터베이스 및 배포  
(개발자가 직접 서버를 구축할 필요 없음)



# 활용 방안

- LSTM 기반 모델인 ELMO로 작동하는 클린봇 프로그램과 같은 역할 수행
- 올바른 게임 문화 및 커뮤니티 문화 형성에 기여하고 게임 산업 전반에 대한 인식 개선



# 제언점

- 문장 단위의 필터링 기능 구현
  - 데이터셋의 개선과 자체적인 labeling 작업 필요
- 데이터셋의 욕설 포함 여부가 하나로 통일되어 있지 않음.
  - 욕설 포함 기준을 통일하는 작업 필요
- 데이터셋의 주기적인 업데이트
  - 최신 욕설 업데이트 작업 필요

**감사합니다!**