Journal of the Korea Industrial Information Systems Research Vol.27 No.2, Apr. 2022 :25-34
http://dx.doi.org/10.9723/jksiis.2022.27.2.025 ISSN:1229-3741

컨볼루션 신경망 모델에 의한 악성 댓글 모자이크처리 방안+

(Blurring of Swear Words in Negative Comments through Convolutional Neural Network)

김 유 민[1], 강 효 빈[2], 한 수 현[2], 정 희 용[2),3)*
(Yumin Kim, Hyobin Kang, Suhyun Han, and Hieyong Jeong)

요 약 온라인 서비스의 발달로 악성 댓글의 파급력이 커져 사이버 폭력 피해가 극심해지고 있다. 이를 방지하기 위해 금칙어 기반 필터링, 신고제도 등 다양한 방법이 사용되고 있지만 악성 댓글을 완벽하게 근절하기는 어렵다. 본 연구는 딥러닝을 사용하여 악성 댓글의 분류의 정확도를 높이고 욕설에 해당하는 부분을 모자이크처리 처리하는 것을 목적으로 진행되었다. 정확도를 높이기 위해 컨볼루션의 층수, 필터 수를 다르게 설정하여 두 가지 모델링을 진행하여 비교하였고, 데이터 세트의 90%를 훈련 데이터로, 10%를 테스트 데이터로 사용한 결과 최종 88%의 정확도를 도출해 낼 수 있었다. 또한 Grad-CAM을 사용하여 모델이 댓글의 어느 부분을 결과에 반영하였는지 표시하여 욕설 위치 정보를 출력하였다. 단순 금칙어 기반으로 댓글을 분류한 정확도는 56%이지만, 컨볼루션 신경망에 의한 분류 정확도가 88%인 것과 비교하면 딥러닝 모델로 악성 댓글의 욕설을 처리하는 것이 더 효과적인 것을 확인할 수 있었다.

핵심주제어: 딥러닝, 악성 댓글, 욕설, 컨볼루션 신경망

Abstract With the development of online services, the ripple effect of negative comments is increasing, and the damage of cyber violence is rising. Various methods such as filtering based on forbidden words and reporting systems prevent this, but it is challenging to eradicate negative comments. Therefore, this study aimed to increase the accuracy of the classification of negative comments using deep learning and blur the parts corresponding to profanity. Two different conditional training helped decide the number of deep learning layers and filters. The accuracy of 88% confirmed with 90% of the dataset for training and 10% for tests. In addition, Grad-CAM enabled us to find and blur the location of swear words in negative comments. Although the accuracy of classifying comments based on simple forbidden words was 56%, it was found that blurring negative comments through the deep learning model was more effective.

Keywords: deep learning, negative comment, swear, convolutional neural network

* Corresponding Author: 정희용(h.jeong@jnu.ac.kr)
+ 이 논문은 2021년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. 2021R1I1A305521011) 연구비와 전남대학교 학술연구비(과제번호: 2021-50561) 지원에 의해 연구되었음.

Manuscript received January 12, 2022 / revised February 12, 2022 / accepted February 28, 2022
1) 전남대학교 공과대학 소프트웨어공학과, 제1저자
2) 전남대학교 공과대학 소프트웨어공학과
3) 전남대학교 인공지능융합학과, 교신저자

1. 서 론

악성 댓글은 인터넷상에서 다른 사람의 명예를 훼손 혹은 비방하거나 모욕을 주는 내용이 담겨 있는 악의적 댓글을 말한다. 줄임말로 악플이라고 표현하는데 이는 한자어 '惡'과 영어 단어 'reply'가 합쳐져서 생성된 합성어이다 (Hong, 2015). 댓글을 통해 제공되는 정보 중 대다수는 기사 의도와 무관하거나 편향된 개인의 의견에 불과하지만, 공격성이 있는 댓글이 달리는 순간 다음 댓글에서도 계속해서 공격적인 성향이 이어지는 현상을 볼 수 있다. 이것을 심리학 용어로 동조라고 지칭하는데 단순히 다른 사람에게 맞춰 행동하게 되는 것을 의미한다. 여기서 야기되는 문제점은 이 동조로 인하여 사이버공간에서 댓글이 여론을 조성하는 효과를 발생시키게 된다는 점이다. 악성 댓글은 상대방 의견에 관한 판단 또는 본인 의지와 상관없이 점점 더 속도와 범위를 늘려가며 확산되어 개인과 사회에 루머를 퍼뜨리며 영향력을 행사하게 된다.

악플을 근절하기 위해 시스템적으로 접근한 연구가 다수 존재한다 (Chen et al, 2012; Nobata et al, 2016; Park et al, 2019; Seo et al, 2017). 악성 댓글을 자주 등록하는 사용자의 IP를 블랙 리스트에 등록하여 댓글 등록을 차단하거나 언어 모델을 이용하는 등 다양한 연구 방법이 제시되어왔다. 하지만 악성 댓글을 작성하는 사용자가 다른 IP를 사용할 때는 같은 문제가 반복적으로 발생하고, 댓글은 단어를 변형하여 쓰는 경우가 많아 언어 모델만을 이용할 경우 악성 단어를 인식하는 데 한계가 있었다. 또한 게임회사에서도 욕설 필터링이나 신고 제도 등 다양한 방지책을 도입하였지만 완벽한 근절로 이어지지는 못하는 것이 현실이다.

악플 탐지 연구는 대부분 영어이며 기계학습 기반으로 진행되었다 (Park et al, 2019). 유튜브의 악성 댓글 탐지, 야후의 금융 및 뉴스 기사를 활용한 욕설 분류를 대표적인 예로 들 수 있다. 금칙어를 정하고 해당 단어를 모자이크처리하는 기존 한국어 악플 방지책은 그 한계가 명확하다. 모음 대신에 숫자를 사용한 변형된 욕설이나, 욕설을 사용하지 않았지만 그 의미가 공격적인 표현은 탐지하기 어렵기 때문이다. 금칙어 기준을 강화하자니 평범한 말도 욕설로 제재되는 현상이 발생하여 사용자에게 오히려 불편함을 안겨주게 된다. 결국 현재는 운영자가 신고가 들어온 내용을 일일이 수작업으로 확인하여 처리해야 하는 방식을 취할 수밖에 없다.

딥러닝 모델을 활용하여 욕설을 탐지하려는 시도는 현재 다양한 분야에서 활발하게 이루어지고 있다 (Kim, Y., 2014; Kim, et al, 2021). 네이버는 클린봇이라는 악성 댓글 탐지 AI로 뉴스, 동영상 댓글을 모니터링하고 있으며 다음 또한 불쾌한 댓글을 자동으로 가려주는 AI인 세이프봇을 댓글난에 적용하고 있다. 온라인 게임 회사 넥슨은 어뷰징 탐지팀을 꾸려 AI로 게임 채팅창의 욕설을 탐지하는 기술을 연구 중이고 영화 추천 애플리케이션 왓챠피디아도 부적절한 콘텐츠 감상평을 가리는 AI 모니터링 기술을 적용 중이다. 이처럼 욕설 탐지 딥 러닝 모델은 현재 포털 사이트, 게임, OTT (Over-the-Top) 등의 서비스에서 연구 중 (Jeong et al, 2021; Kim et al, 2018; Ryu et al 2020)이며 앞으로도 이에 관한 연구는 더 다양한 플랫폼에서 이뤄질 것으로 보인다.

이에 본 연구에서는 인터넷상에서 작성되는 방대한 양의 데이터 사이에서 악의적 데이터를 정확히 탐지하기 위해 크롤링(crawling)으로 모은 데이터 속 단어를 형태소 단위로 토큰화한 다음, 컨볼루션 신경망 모델(CNN, Convolutional Neural Network Model)로 학습시켜 악성 댓글을 잘 분류할 수 있는지 확인하도록 하겠다. 또한 악성 댓글의 소지가 있는 단어가 문장의 어느 위치에 포함되어 있는지 표시하여 모자이크처리 할 수 있는지도 살펴보도록 하겠다. 이 기능은 딥러닝의 예측결과가 적절한지 어떤지를 모자이크처리된 부분이 욕설에 해당하는지 아닌지로 알 수 있기 때문에 딥러닝 모델을 평가할 수 있다.

2. 데이터셋 구축

	A	B	C	D	E	F
1		id	label	data		
2	1	1		돼지유전자 멸족		
3	2	2		헐... 진짜 작살났네		
4	3	3		미국에서 태어난 희다ㅈㅈㅈ		
5	4	4		애들 마블 나이키사려고 알바비 한 푼 두 푼 모으는 새끼가ㅋㅋ미국 한 번 가보고 싶은 마음을 그렇게 폭화시키누		
6	5	5		상금이님은 알바도 안 하고 엄마 돈 빼앗아서 닌텐도 사신다		
7	6	6		상금이 여기도 있너 - dc App		
8	7	7		참고 건강한 사람중에 일부는 그냥 모르고 넘어갈 수 있다는거지, 괜찮은건 아니다.		
9	8	8		강해보이는데 바이러스에는 노답이구만		
10	9	9		대부분 비만 환자네.. 당뇨나 지병 100% 있을듯		
11	10	10		아무리봐도 코로나바이러스는 서양인의 dna 에 맞춰서 개발된거같음 동남아나 아시아인에비해 치사율이 너무높다 이쯤되면 짱깨+러시아가		
12	11	11		러시아가 동양국가노		
13	12	12		박쥐나 천산갑에서 그랬다는건 개구라지 짱개가 하루이틀 박쥐 천산갑 잡아먹은것도 아닐테고 말여 사실상 짱개의 화생방 테러 - dc App		
14	13	13		러시아 아시아 맞는데 통천지 ㅂㅅ인가?		
15	14	14		유럽 아시아 다 걸친다 병신아 - dc App		

Fig. 1 Dataset for labelling.

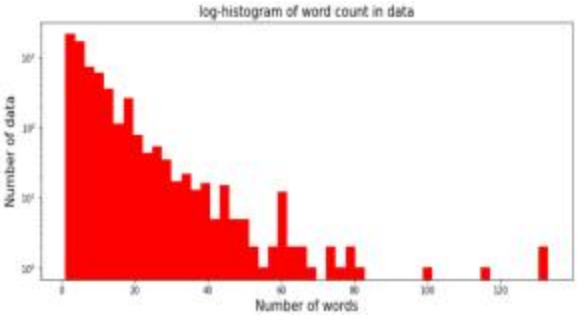

Fig. 2 Distribution of words in all comments.

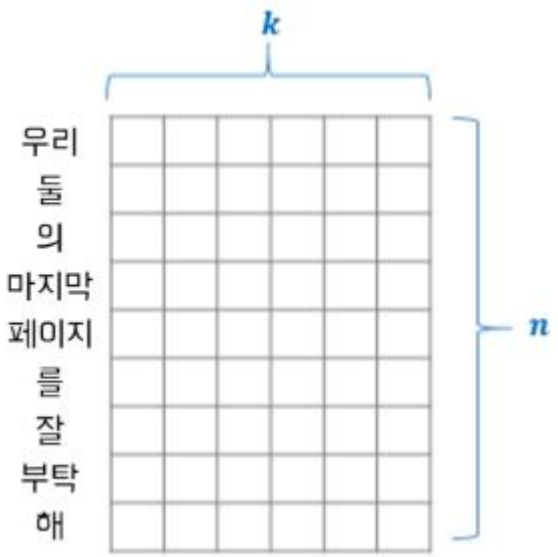

Fig. 3 Results of sentence vector after word embedding.

연구의 시작으로 우리는 먼저 오픈 커뮤니티인 디시인사이드의 댓글을 2021년 4월 1일부터 일주일간 수집하였다. 댓글 수집 사이트로 디시인사이드를 선정한 이유는 네이버(Naver), 다음(Daum)의 포털 사이트들은 이미 댓글 난에 필터링 기능을 적용하고 있어 악성 댓글의 비율이 적기 때문이다. 반면 디시인사이드는 필터링이 적용되지 않고 회원가입 없이 누구나 댓글을 쓸 수 있다는 점에서 접근성이 좋아 다양한 유형의 댓글을 찾아볼 수 있었다. HTML에서 원하는 태그를 추출할 수 있는 파이썬 라이브러리 BeautifulSoup를 사용하여 디시인사이드 이슈 카테고리의 게시글의 댓글을 크롤링하여 텍스트 파일로 저장했다. 댓글은 별도의 기준을 두지 않고 모두 가져오도록 했으며 댓글 하나를 수집할 때마다 텍스트 파일 뒤에 붙여넣는 방식으로 총 7,094개의 댓글을 확보했으며 Fig. 1에서 보여주고 있듯이, 욕설이 포함된 댓글은 0으로, 욕설이 포함되지 않은 댓글은 1로 라벨링 하였다. 댓글 데이터의 라벨은 0과 1이 약 1:2의 비율로 분포되어 있는 것을 확인하였다.

훈련 데이터와 테스트 데이터는 9:1의 비율로 분리했다. 훈련 데이터의 수는 6,205개로, Fig. 2는 각각의 훈련 데이터 내 단어 수 분포를 그래프로 표시한 것이다. 더 구체적인 수치를 출력한 결과, 댓글 내 단어 수는 평균 8개였고 최대 단어 수는 133개로 나타났다. 이 수치는 추후 데이터 전처리 과정에서 패딩(padding) 처리 시 사용하게 된다.

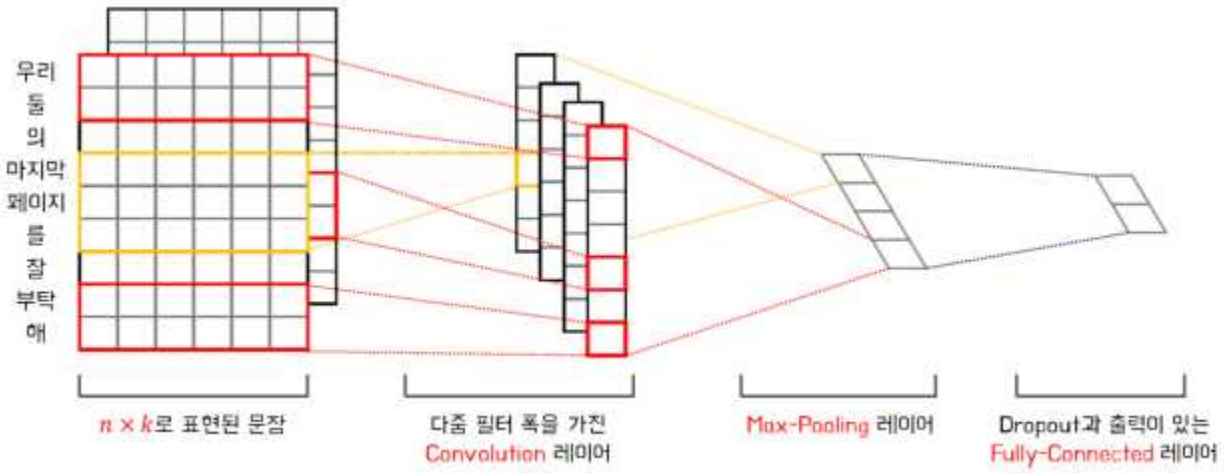

Fig. 4 One-dimensional convolutional neural network.

```
# Okt() 테스트
okt.morphs(sample_text, stem=True)
```

```
['어제', '먹다', '라면', '때문', '에', '배', '가', '아프다']
```

Fig. 5 Results of Korean morphological analyzer of OKT.

3. 컨볼루션 신경망 모델

3.1 모델 요구사항

본 연구에서 필요한 요구사항은 세 가지로 요약할 수 있다. 첫 번째는 높은 분류 정확도를 목표로 최소 56%이상의 정확도를 도출해야 한다. 이는 금칙어 기반 분류 정확도보다 높은 수치를 보여야 하기 때문이다(Cho, Y., 2018). 두 번째는 댓글 내 욕설의 위치를 파악할 수 있어야 한다. 욕설 탐지 뿐만 아니라 그 위치까지 표시함으로써 컨볼루션 신경망 모델의 우수성을 확인해 볼 수 있다. 이 기능은 딥러닝의 학습 결과를 설명할 때 사용되어질 수 있다. 즉, 바른 예측 결과라고 하더라도 그 근거가 무엇인지를 모델 자체로는 확인해 볼 수 없기 때문에 딥러닝의 예측결과에 대한 판단 기준이 적절한지 확인해 볼 수 있다. 마지막은 변형된 욕설이나 단어 사이에 글자를 끼워 넣은 욕설의 경우라도 데이터셋에 포함시키면 분류가 가능해야 한다. 이 요구사항 또한 컨볼루션 신경망 모델의 우수성을 보이기 위해 충족되어야 한다.

3.2 모델 설계

본 연구에서 사용한 신경망은 컨볼루션 신경망(CNN)이다. CNN은 이미지 처리에서 주로 사용하는 신경망으로 가중치를 가진 필터가 특성을 추출해 나가며 데이터를 학습하는 모델이다. 우리는 이 신경망을 사용해 댓글에서 특성, 즉 토큰을 추출하고 모델을 학습시킨 뒤 욕설 포함 여부를 분류하는 것으로 전체 과정을 설계하였다.

텍스트 데이터는 3차원인 이미지와 달리 2차원 벡터로 구성되기 때문에 1차원 CNN을 사용한다. 1차원 CNN의 초기 입력은 컴퓨터가 이해할 수 있는 형태로 벡터화된 행렬인데, 이 행렬을 만드는 과정을 워드 임베딩(word embedding)이라고 한다. Fig. 3은 워드 임베딩(word embedding) 후 문장이 어떻게 벡터로 변환되는지를 보여주며 이때 n은 벡터 차원, k는 댓글 길이를 나타낸다. 이 2차원 벡터는 가중치를 가진 커널(kernel)과 결합하여 합성곱 연산을 수행하고 특성 맵이라 불리는 1차원 벡터를 생성한다. 각각 다른 크기의 커널(kernel)로 만들어진 특성 맵들에서 가장 큰 값만을 빼내는 맥스 풀

Fig. 6 Word dictionary displaying words associated with the sequence index.

링(max pulling) 연산을 수행하면 문장에서 중요한 특성을 추출할 수 있다.

Fig. 4는 지금까지의 과정을 하나로 연결한 것이다. 문장의 임베딩(embedding) 벡터가 CNN 층에서 합성곱 연산을 수행하고 특성 맵들을 만들면 맥스 풀링(max pulling) 층에서 가장 큰 값들을 연결하여 하나의 벡터로 만든다. 이 벡터는 FC(fully connected) 층에서 뉴런이 1개인 출력층을 지나 최종적으로 이진 분류를 수행하게 된다.

4. 결 과

4.1 데이터 전처리 결과

댓글 데이터의 전처리 결과이다. 우선 훈련 데이터와 테스트 데이터를 각각 형태소 단위로 토큰화하는 작업을 진행했는데 이때 작업 모듈은 형태소 분석기 KoNLPy의 OKT(Open Korean Text)를 사용했다. KoNLPy는 한국어를 처리하기 위한 다양한 기능을 제공하는 파이썬 패키지로, OKT는 KoNLPy에 내장된 형태소 분석기이다. OKT는 Fig. 5와 같이 문장이 들어왔을 때 형태소 단위로 분리된 문장의 리스트를 반환한다.

한글과 공백을 제외한 모든 문자를 제거한 뒤 OKT를 사용하여 토큰화하였고 의미를 갖지 않는 불용어들을 제거했다. 이때 제거한 불용어에는 '은', '는', '이', '가', '에게'와 같은 조사와 '등', '지', '한', '하다'와 같이 의미상으로 중요하지 않은 단어들이 포함된다. 이렇게 만들어진 리스트는 정수 시퀀스로 변환하였고 각 정수와 연관된 토큰을 표시하는 단어 사전도 함께 생성했다. 이후 시퀀스들의 길이가 같아지도록 최대 단어 수인 133으로 패딩 처리를 진행했다. Fig. 6은 시퀀스 인덱스와 연관된 단어를 표시하는 단어 사전의 결과를 보여주고 있다.

4.2 학습 모델 결정

우리는 정확도 향상을 위해 본 연구에서 총 두 차례의 모델링을 수행했다. 첫 번째 모델링은 시퀀스(Sequential) 함수를 사용하여 설계했다. 시퀀스(Sequential) 함수는 입력층, 은닉층, 출력층을 구성하며 add 함수를 통해 층을 추가하도록 되어있다. 손실률이 증가하면 과적합이 일어나고 있다는 뜻이므로 조기종료(EarlyStopping) 기능을 통해 손실률이 세 번 증가하면 학습을 종료하도록 하였다. Fig. 7은 최종적으로 사용한 모델의 구조이다. 밀집층에서 ReLU를, 출력층에서 Sigmoid를 활성화 함수로 사용했다. 테스트 결과, 정확도는 0.869로 이전 모델보다 향상된 것으로 나타났다. Fig. 8에서는 에폭(epoch)에 따른 모델의 정확도와 손실률 그래프이다. 에폭(epoch) 10 이후로는 과적합(overfitting)이 발생하고 있으므로 실제 학습은 에폭(epoch) 10에서 종료한 것을 사용하였다.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 256)	2300160
dropout (Dropout)	(None, None, 256)	0
conv1d (Conv1D)	(None, None, 32)	24608
global_max_pooling1d (Global	(None, 32)	0
dense (Dense)	(None, 250)	8250
dropout_1 (Dropout)	(None, 250)	0
dense_1 (Dense)	(None, 1)	251

Total params: 2,333,269
Trainable params: 2,333,269
Non-trainable params: 0

Fig. 7 The first sequential model.

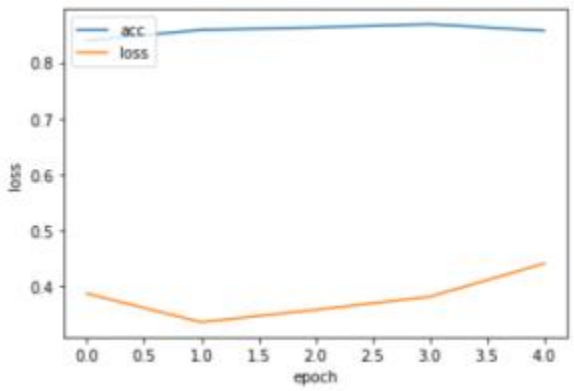

Fig. 8 The plot of the training for the first sequential model.

여러 차례의 테스트 후 두 번째 모델링에서는 총 세 개의 컨볼루션 층을 사용했다. Fig. 9는 두 번째 모델의 결과를 보여주고 있다. 순서대로 필터 폭이 6, 3, 2인 컨볼루션 층과 풀링 층을 지나 뉴런이 1개인 출력층을 통해 마지막으로 예측 결과를 출력했다. 이렇게 설계한 두 번째 모델의 정확도는 0.878로 나타났고 이는 지금까지의 모델 정확도 중에서 가장 높은 수치였다. Fig. 10의 그래프는 이 모델의 에폭(epoch)에 따른 정확도와 손실률을 보여주고 있다. 두 번째 모델의 경우에서도 에폭(epoch) 10 이후로 과적합(overfitting)이 발생하고 있으므로 실제 학습은 에폭(epoch) 10에서 종료한 것을 사용하였다.

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 133)]	0
embedding (Embedding)	(None, 133, 64)	573568
conv1d (Conv1D)	(None, 128, 64)	24640
max_pooling1d (MaxPooling1D)	(None, 64, 64)	0
conv1d_1 (Conv1D)	(None, 61, 64)	16448
conv1d_2 (Conv1D)	(None, 60, 64)	8256
global_max_pooling1d (Global	(None, 64)	0
dense (Dense)	(None, 32)	2080
preds (Dense)	(None, 1)	33

Total params: 625,025
Trainable params: 625,025
Non-trainable params: 0

Fig. 9 The second sequential model.

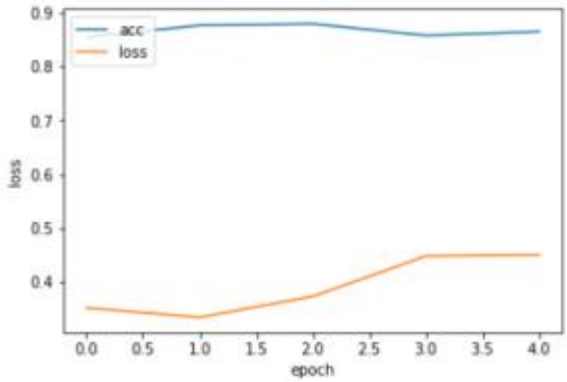

Fig. 10 The plot of the training for the second sequential model.

```
index = int(input())
prob = model.predict(X_test)

prob[index], test_data.iloc[index], y_test[index]
```

```
80

(array([0.02488706], dtype=float32),
 id                                6591
 label                                0
 data     저거 내용만봐선 씨발 무슨 스토리인지 모르자나
 Name: 6590, dtype: object,
 0)
```

Fig. 11 The classification result.

```
norm_len = MAX_SEQUENCE_LENGTH / last_conv_layer.output_shape[1]

html = ""

choose_index = int(input())

if y_pred[choose_index] > 0.5:
  pred = 'does not include'
else:
  pred = 'includes'

html += "<span><h3>This comment {} abusive language.".format(pred)
html += "</h3></span><br>"

for j, i in enumerate(tokenizer.sequences_to_texts(X_test)[choose_index].split()):
  html += "<span style='background-color:rgba({},0,150,{})'>{} </span>".format(heatmap[math.floor(j / norm_len)] * 255,
                                                                               heatmap[math.floor(j / norm_len)] - 0.4, i)

HTML(html)
```

87

This comment includes abusive language.

진짜 ㅈ 같이 판매 애 밉다 뒤지다 봄 카드 지갑 만원 이나 해처 먹음

This comment includes abusive language.

닥치다 씨발련

Fig. 12 The heat map results through Grad-CAM.

This comment does not include abusive language.

게임 참 ㅅ1111ㅂ 같이 하네

Fig. 13 An example for the failed results.

4.3 모델 평가

두 번째 모델을 최종 모델로 선정하여 이 모델이 악성 댓글을 잘 분류하고 있는지 몇 가지 댓글을 통해 확인해보았다. 댓글의 라벨과 마지막 정수 출력을 비교해 본 결과, 욕설이 포함된 댓글을 0으로 잘 분류했음을 알 수 있었다. Fig. 11는 실제 라벨이 0인 6591번째 댓글이 모델에 의해 0.024로 예측된 모습이다. 0.024는 0에 가까우며 이는 모델이 이 댓글을 욕설을 포함하고 있는 악성 댓글로 잘 분류한 것을 나타낸다.

다음은 학습된 내용을 바탕으로 모델이 결과에 많이 반영한 토큰을 표시함으로써 댓글 내 욕설 위치를 출력했다. Fig. 12은 중요한 토큰을 히트맵(heatmap)으로 표시한 결과를 보여주고 있다.

히트맵(heatmap)으로 Grad-CAM(Nguyen, et al, 2021; Selvaraju, et al, 2019)을 구현해 확인한 결과, 변형된 욕설일 지라도 학습시킨 모델이 그 위치를 잘 파악하고 있음을 확인하였다. 따라서 데이터셋에 포함만 되어 있다면 CNN이 욕설을 바르게 탐지하고 있다는 것을 알 수 있다. 하지만, Fig. 13에서 “게임 참 ㅅ1111ㅂ 같이 하네”는 데이터셋에 없는 단어로 “ㅅ1111ㅂ”은 변형된 욕설임에도 불구하고 데이터셋에 포함되지 않았기 때문에 잘못된 예측결과를 확인할 수 있다.

따라서, 본 모델은 데이터셋에 포함되어 있다면 변형된 욕설일지라도 바르게 예측을 할 수 있지만, 데이터셋에 포함되어 있지 않다면 예측결과가 바르지 못한 것을 알 수 있다.

	A	B
1		keyword
2	0	긴나
3	1	개긴나
4	2	쌈긴나
5	3	즘긴나
6	4	좃긴나
7	5	갑보
8	6	개긴년
9	7	개나리
10	8	개년
11	9	개돼지
12	10	개새끼
13	11	개좃
14	12	개쓰레기
15	13	개소리
16	14	개씨발

Fig. 14 The part of the slang dictionary.

5. 토 의

Fig. 14는 금칙어 사전의 일부를 보여주고 있다. 한국의 욕설 관련 도서로 우리말 상소리 사전(정태룡, 프리미엄 북스, 1994), 한국의 욕설백과(정태룡, 한국문원, 1997), 토속어 성속어 사전(정태룡, 우석, 2000), 농어속담사전(송재선, 동문선, 1995), 여성속담사전(송재선, 동문선, 1995), 상말속담사전(송재선, 동문선, 1993)이 있다, 또한 게임산업진흥원에서 펴낸 '게임언어 건전화 지침서 연구' 보고서에 포함된 금칙어 데이터베이스도 활용할 수 있다(Korea Game Industry Agency, 2008). Fig. 15에서는 욕설 데이터 세트를 사용하여 단순 금칙어 기반으로 댓글을 분류한 정확도이다. 분류에는 온라인 상에 배포되어 있는 비속어 사전을 사용하였으며 댓글 데이터에 해당 단어가 포함되어 있으면 예측

```
print("₩n 정확도: %.4f" % ((6895 - sum) / 6895))
```

```
정확도: 0.7131
```

Fig. 15 The classification results based on forbidden words.

라벨을 0으로 지정했다. 이후 예측한 라벨과 실제 라벨이 일치하는 댓글의 수를 백분율로 표시하여 정확도를 계산했다. CNN의 분류 정확도가 0.878이었던 것과 비교하면 금칙어 기반으로 제안한 CNN 모델을 학습시킨 후 예측한 데이터셋의 분류 정확도는 상대적으로 낮다는 것을 알 수 있다. 이를 통해 딥러닝 기반 욕설 탐지가 단순 욕설을 필터링하는 것보다 성능이 더 좋다는 것을 증명할 수 있었다.

본 연구의 목적은 현재 널리 사용되고 있는 금칙어 기반 필터링 방법과 신고 제도의 한계를 인식하고 더 정확하게 악성 댓글을 분류할 수 있는 딥러닝 모델을 설계하는 것이었다. 금칙어 기반 분류 정확도와 딥러닝 모델의 분류 정확도를 비교해보면 이는 기존 방식의 문제점을 보완할 수 있는 효과적인 방안임을 알 수 있다. 하지만, 본 연구에 제안한 방법으로는 소셜네트워크(SNS) 운영자가 변형된 욕설이나 새로운 욕설이 나올 때 마다 데이터셋에 입력하고 재학습을 시켜야 한다는 한계점이 있기 때문에 컴퓨터가 욕설을 자동으로 인식시키기 위한 새로운 방법이 필요해 보인다.

6. 결 론

본 연구에서는 1차원 컨볼루션 신경망을 이용하여 욕설이 포함된 악성 댓글을 효과적으로 탐지할 수 있음과 히트맵(heatmap)으로 댓글 내 욕설 위치 정보를 표시할 수 있음을 확인했다. 현재 욕설 탐지에 널리 이용되고 있는 금칙어 필터링 시스템이나 신고 제도로는 완벽한 욕설 근절이 어렵다. 이에 딥러닝 활용은 효과적인 필터링 결과를 제공하며 이를 소셜네트워크(SNS)나 온라인 게임에 적용한다면 사이버 폭

력 감소에 직접적인 도움이 될 수 있을 것으로 생각한다.

연구의 한계점 또한 존재한다. 본 연구에서는 하나의 온라인 커뮤니티 댓글만을 사용했다. 커뮤니티나 SNS 별로 사용하는 은어가 다르므로 다양한 서비스에서 더 많은 댓글을 수집하여 데이터 세트로 사용한다면 모델의 성능이 향상될 수 있을 것으로 보인다. 또한 기존 데이터셋에 없지만 변형된 신조 욕설을 탐지하여 자동으로 데이터셋에 포함시키는 기능은 추후 보완해야 할 점이다.

Acknowledges

본 논문은 김유민, 강효빈, 한수현의 2021년도 학사 학위 논문에서 발췌 정리하였음.

References

Chen, Y., Shou, Y., Zhu, S. and Xu, H. (2012). Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. In Privacy, Security, Risk and Trust, IEEE, pp. 71-80.

Cho. Y. (2018). Detecting swear words through deep learning in NDC2018 (http://ndc.vod.nexoncdn.co.kr/NDC2018/slides/NDC2018_0033/inde x.html).

Jeong, H., Ko, J. and Shin, C. (2021). Abnormal Detection with Microscope through Deep Learning. Journal of the Korean Industrial Information Systems Research, 26(2), pp. 1-10.

Kim, J., Jo, H. and Kim, B. (2018). Game Recommendation System Based on User Ratings. Journal of the Korean Industrial Information Systems Research, 23(6), pp. 9-19.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751.

Kim, Y., Kang, H., Han, S. and Jeong, H. (2021). Swear Word Detection through Convolutional Neural Network, Proceedings of the Korea Information Processing Society Conference, pp. 685-686.

Korea Game Industry Agency. (2008). Study on game language guidelines (https://www.korean.go.kr/attachFile/viewer/202202/3cc1548a-ef43-4043-9c5c-af95e3d39559.pdf.htm),(https://www.korean.go.kr/front/etcData/etcDataView.do?mn_id=208&etc_seq=121).

Nguyen, T.P.H., Shin, C. and Jeong, H. (2021). Finding the Difference in Capillaries of Taste Buds between Smokers and Non-Smokers Using the Convolutional Neural Networks. Applied Sciences, 11(8). 3460 (https://doi.org/10.3390/app11083460).

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y. and Chang, Y. (2016). Abusive Language Detection in Online User Content. In WWW, pp. 145-153.

Park, S., Kim, H. and Woo, J. (2019). Abusive Setence Detection using Deep Learning in Online Game. Proceedings of the Korean Society of Computer Information Conference, 27(2), pp. 13-14.

Ryu, M. and Cho, H. (2020). An Analysis of IoT Service using Sentiment Analysis on Online Reviews: Focusing on the Characteristics of Service Providers. Journal of the Korean Industrial Information Systems Research, 25(5), pp. 91-102.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision, 128, pp. 336-359.

Seo, S. and Cho, S. (2017). Transfer Learning

Method for Solving Imbalance Data of Abusive Sentence Classification. Journal of KIISE, 44(12), pp. 1275-1281.

Hong, Jinju. (2015). A malicious comments detection technique on the internet. Master Thesis, Soongsil University.

김 유 민 (Yumin Kim)

- 전남대학교 소프트웨어공학과 학사
- 관심분야: 딥러닝, 클라우드 컴퓨팅, 자연어처리

강 효 빈 (Hyobin Kang)

- 전남대학교 소프트웨어공학과 학사
- 관심분야: 딥러닝, 자연어처리

한 수 현 (Suhyun Han)

- 전남대학교 소프트웨어공학과 학사
- 관심분야: 딥러닝, 자연어처리

정 희 용 (Hieyong Jeong)

- 정회원
- 부경대학교 제어계측공학과 학사
- Hiroshima University 로봇공학 석사
- Osaka University 기계공학 박사
- (현재) 전남대학교 AI융합대학 인공지능융합학과 교수
- 관심분야: 헬스케어, 지능로봇, 인간-로봇 상호작용, 컴퓨터비전, 자연어처리, 음성처리