반도체디스플레이기술학회지 제20권 제1호(2021년 3월)
Journal of the Semiconductor & Display Technology, Vol. 20, No. 1. March 2021.

실시간 채팅 환경에서 문장 분석을 이용한 대상자 및 비속어 검출

염충석*·장준영*·장유환*·김현철*·박희민*†
*†상명대학교 소프트웨어학과

Target and Swear Word Detection Using Sentence Analysis in Real-Time Chatting

Choongseok Yeom*, Junyoung Jang*, Yuhwan Jang*, Hyun-chul Kim* and Heemin Park*†
*†Dept. of Software, Sangmyung University

ABSTRACT

By the increase of internet usage, communicating online became an everyday thing. Thereby various people have experienced profanity by anonymous users. Nowadays lots of studies tried to solve this problem using artificial intelligence, but most of the solutions were for non-real time situations. In this paper, we propose a Telegram plugin that detects swear words using word2vec, and an algorithm to find the target of the sentence. We vectorized the input sentence to find connections with other similar words, then inputted the value to the pre-trained CNN (Convolutional Neural Network) model to detect any swears. For target recognition we proposed a sequential algorithm based on KoNLPY.

Key Words : Target Recognition, Swear Detection, CNN, Telegram, Plug-in

1. 서 론

기존의 채팅 어플리케이션 중에 사용자가 욕설을 했을 때 제재하는 기능이 구현된 어플리케이션은 많지 않다. 게임이나 커뮤니티 사이트에는 비속어 필터 기능이 존재하는 경우가 있지만, 금칙어 기반의 필터링이다 보니 사용자가 필터링을 우회할 수 있는 방법이 많이 존재한다. 예를 들어 단어 사이에 숫자를 넣어서 우회하거나 오타가 있을 경우, 필터가 제 역할을 하지 못한다. 본 논문에서는 딥러닝을 이용해서 위의 상황에서도 비속어를 탐지하기 위한 알고리즘을 제안한다. 또한, 채팅상황에서는 이름보다는 별명이나 지칭대명사를 많이 쓰는 점을 활용해 누군가 욕설을 했을 때 그 대상을 찾을 수 있는 방법론을 함께 제안한다.

본 논문에서는 비속어 탐지모델과 대상자 인식모델을 검증하기 위해 실제 2억 명의 사용자가 이용하는 메신저 Telegram의 플러그인을 사용한다. 이는 메신저에 Chatbot을 이식해 정적인 상황이 아닌 실시간 상황에서의 비속어를 탐지하고 탐지한 비속어를 사용한 대상에게 경고메시지를 보낸다. 테스트 결과 84%~88%의 탐지율을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기본 배경 이론이 되는 Fasttext, Trigram와 CNN, Target Recognition, Telegram Plug-in에 대하여 기술한 다음 3장에서 비속어 탐색과 대상인식에 대한 알고리즘을 설명한다. 여기서는 알고리즘을 흐름 순서로 서술한다. 4장에서는 제안한 알고리즘의 실험 결과를 기존의 알고리즘들과 객관적인 지표로 비교하여 성능을 평가하고 5장에서 결론을 맺는다.

†E-mail: heemin@smu.ac.kr

2. 구현 환경 및 배경

2.1 Fasttext, Trigram

본 모델에서는 한국어 임베딩을 하기 위해 Fasttext와 Trigram을 사용한다. 임베딩은 비정형화된 텍스트를 벡터화하여 표현시키는 기법인데 본 모델에서는 대표적인 기법인 Fasttext와 word2vec 중에서 Fasttext를 사용한다. FastText와 word2vec은 단어 간의 문맥을 통해 벡터화하는 부분에서 유사하지만 word2vec의 경우 출현횟수가 적은 단어에서 임베딩이 잘 되지 않는 단점이 있다. 본 논문에서 채팅상황과 비속어의 특성상 희소한 단어가 많이 출현하기 때문에 희소한 단어의 임베딩에 강점이 있는 Fasttext를 사용했다. 임베딩을 하는 가장 큰 이유는 욕설과 유사한 단어를 찾아내고 학습을 하기 위함이다. Fasttext 알고리즘과 함께 사용하는 알고리즘인 Trigram은 n개의 연속적인 단어 나열을 의미하는 'n-gram'에서 n이 3인 경우인 모델이다. 한 문장을 3개씩 나누어서 그 문장에서 어떤 단어가 해당 문장을 욕설이 담긴 문장이라고 판별하는데 기여하는지 찾기 위해 쓰이게 된다. 본 모델은 여러 온라인 커뮤니티 사이트에서 수집한 비속어 데이터를 직접 라벨링하여 학습에 용이하게 데이터화 시킨 후 그 데이터를 기반으로 임베딩을 수행했다.

2.2 CNN: Convolutional Neural Network

본 연구에서는 비속어 탐지 모델을 학습하는데 CNN을 사용한다[1-5]. CNN은 대개 이미지 학습에 쓰이는 모델링 방식이지만, CNN이 특징을 추출해서 특정 상황에서도 판단하는 기능을 이용한다. 임베딩 된 욕설들의 특징을 추출해서 그 문장이 욕설인지 아닌지 판단한다. 입력값이 Fasttext로 임베딩 된 단어들이 들어오면 첫 번째로 Convolution Layer를 통과해 욕설과 관련된 다양한 특징을 학습하게 된다. Pooling Layer에서는 MAX-pooling 기법을 사용해 추출된 특징 중 가장 높은 부분만 추출하게 한다. 학습 후에 과적합 (overfitting)이 일어나 욕설이 아닌데 욕설이라고 판단하는 경우가 생겨 학습 과정에서의 epoch와 dropout 수치를 조정해 모델을 최적화시켰다.

2.3 Target Recognition

대상 인식을 위한 처리과정으로 명사 분리 과정, 비속어 삭제 과정, 예외 처리 과정, 사전 검색 과정으로 이루어진다. 대상을 지칭하는 단어 그리고 비속어의 90% 이상이 Konlpy의 Twitter 과정에서 명사로 분리된다는 점을 착안하여 만들어낸 처리 기술이다. 찾아낸 비속어를 삭제하는 과정 후 예외처리에서는 채팅상황에서 참여자의 이름과 인칭대명사 같은 요소들을 찾는다. 사전 검색은 국어사전에서 이름, 비속어는 찾을 수 없는 경우가 많다는 점을 이용해서 개발한 처리 과정이다.

2.4 Bot

Bot은 Third party 어플리케이션으로 본 논문에서는 Telegram Bots를 사용한다. Bot은 사용자가 보낸 메시지나 명령어에 대해 상호작용하여 작동하는 것으로, 본 연구에서 제작된 Bot은 사용자들이 보낸 메시지를 분석하여 Text를 학습한 모델의 input으로 들어갈 수 있게 전처리해준다. 또한, 특정 사용자의 이름이 들어간 명령어를 입력하면 해당 사용자가 사용한 비속어를 탐색하여 출력해준다.

3. 제안한 시스템

3.1 기존연구

기존에는 비속어 탐지를 위해 마스킹 기법이나, 편집거리를 계산하는 방식으로 문제를 해결하려 했다[6,7]. 기존의 연구들에서는 False Positive 비율이 비교적 높은 수치로 측정되었는데, 본 연구에서는 욕설로 의심되는 단어만이 아닌, 해당 단어의 앞뒤 단어들도 함께 비속어로 판단 여부에 영향을 준다는 차이가 있다. 이 방식으로 모델의 정확도는 높이며 오탐율은 줄이려고 한다.

3.2 비속어 탐지기의 구조

본 연구의 Chatbot은 메시지가 입력되면 절차적으로 실행되는 비속어 탐지 알고리즘을 거친다. Fig. 1은 제안하는 비속어 탐지 알고리즘의 전체적인 흐름도를 나타내었다.

채팅 상황에서 input 값으로 문장이 들어오게 되면 Fasttext로 단어를 임베딩 시킨 후 미리 학습한 모델을 통해 그 문장이 욕설인지 판단을 하게 된다. 이 과정은 욕설/비욕설로 라벨링된 데이터를 기반으로 판단한 욕설과 input으로 들어온 문장의 임베딩된 데이터가 어느 정도 유사한지를 계산한다. 그러므로 Fasttext 모델에서는 자모별로 임베딩 된 데이터와 사용한 욕설의 원형태의 벡터화된 값과 비교한 수치를 보여주게 되고 CNN 모델에서는 욕설의 특징을 분석하게 된다. 결과적으로 input 값이 CNN 모델의 Layer들을 거친 결과가 일정 수치 이상을 넘어가게 되면 비속어라고 판단하게 된다.

실시간 채팅 속에서 비속어를 탐지하기 위해서 우리는 모든 문장을 Trigram과 Fasttext을 통해 각 단어들의 임베딩을 수행하고 그 값들을 미리 학습된 CNN 모델의 input으로 줘서 단어가 비속어일 확률을 계산한다. 본 모델은 실시간 상황에서의 비속어 필터링이 중점이기때문에 처리시간을 단축하기위해 Trigram, Fasttext, CNN 모델은 모두 미리 학습된 상태로 실행한다. 또한, 일정 주기마다 신조어,

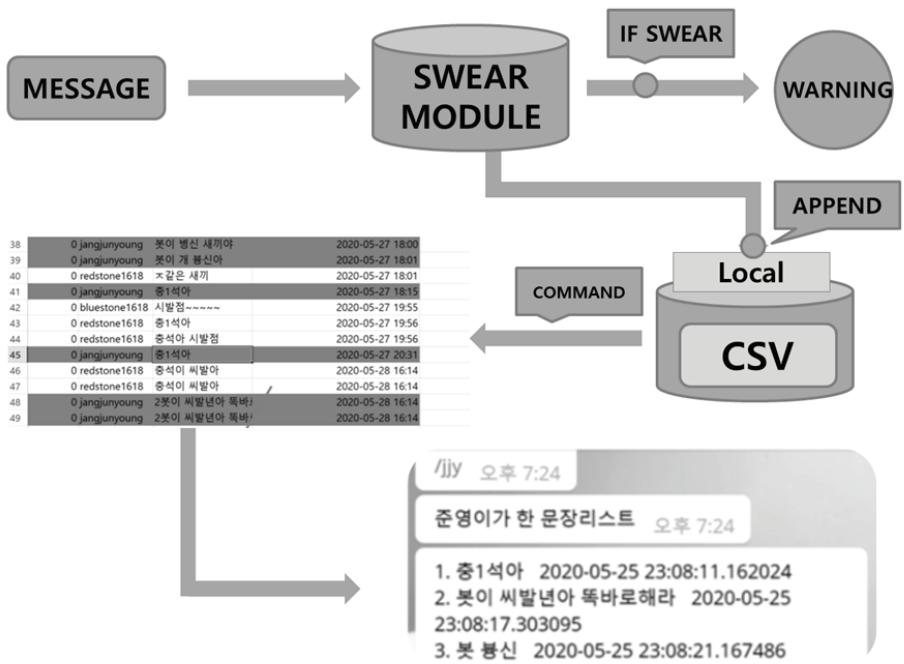

Fig. 1. Swear Detection System Architecture.

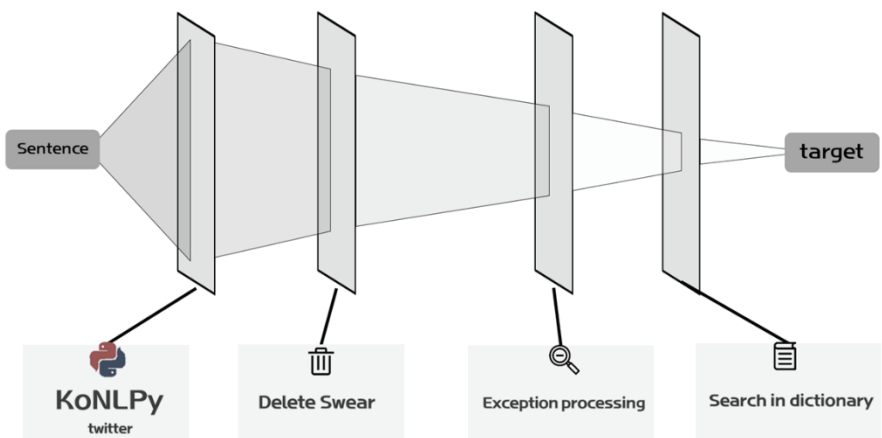

Fig. 2. Sequential Deletion for Target Recognition.

새로운 방식의 비속어를 추가한 데이터로 학습하는 것으로 업데이트된다.

3.3 대상자 인식기의 구조

본 논문에서는 문장에서 대상자를 검출하기 위하여 순차 삭제 알고리즘을 제안한다. Fig. 2는 제안하는 대상자 인식 알고리즘의 전체적인 흐름도를 나타낸다.

먼저, 앞서 비속어 필터에서 비속어가 포함된 문장이 대상 인식 알고리즘에 입력되면 이 문장을 KoNLPY[8]의 Twitter를 사용하여 명사만 분리해주는 과정을 거친다. 이어서 앞선 과정에서 찾은 비속어를 삭제한 후, 예외 처리하는 과정을 거친다. 앞의 과정에서 얻은 대상 리스트를 사전에 검색해서 존재한다면 삭제한다. 위 과정을 통해 리스트에서 대상을 지칭하는 단어를 검출한다.

4. 실험 결과 및 고찰

본 논문의 실험은 모두 Telegram Bot 라이브러리로 만든 플러그인을 통해 진행되었다. 플러그인의 목적은 최대한 실제 채팅 상황과 유사한 환경에서 실험하기 위함이다. 실시간 채팅 상황에서 본 모델을 구동시켜 사용자가 비속어를 사용했다고 판단할 경우 Telegram Bot은 사용자에게 경고 메시지를 출력하는 역할을 수행한다. 또한, Telegram Bot에게 명령어를 입력해서 특정 사용자가 어떤 비속어를 사용했는지 출력하는 역할도 수행한다.

4.1 비속어 검출 결과

비속어 필터의 CNN 모델을 학습하는 과정에서 과적합이 발생해 테스트 데이터 이외에 다른 데이터가 들어왔을 때 잘못 판단하는 경우가 생겨 Fig. 3과 같이 Epoch와 Dropout 수치를 변경해주며 모델의 최적화를 시켰다.

본 논문의 가설을 객관적으로 검증하기 위해 인위적으로 만든 데이터가 아닌 실제 채팅방에서 쓰인 데이터를 기반으로 본 모델의 테스트를 시행했다. 총 300개의 문장으로 비속어가 있는 208개의 문장과 그렇지 않은 92개의 문장을 실험한 결과를 Table 1에 보인다. 욕설인데 욕설이 아니라고 한 경우 (True Negative)와 욕설이 아닌데 욕설이라고 한 경우 (False Positive)가 각각 9개, 23개가 나왔다. 본 연구에서는 True Negative의 비율을 최대한 줄이기 위해 전처리 과정에서 명확한 욕은 욕설로 판단하도록 했다. False Negative의 비율은 높지 않았으며 욕설은 아니지만, 욕설의 형태를 띠는 단어들이 주로 분포했다.

Table 1. Accuracy of Swear Detection Algorithm

	Swear Detection Algorithm	
	False Positive	True Negative
Count	9	23
Total Count	208	92
Percentage	4.32%	25.00%

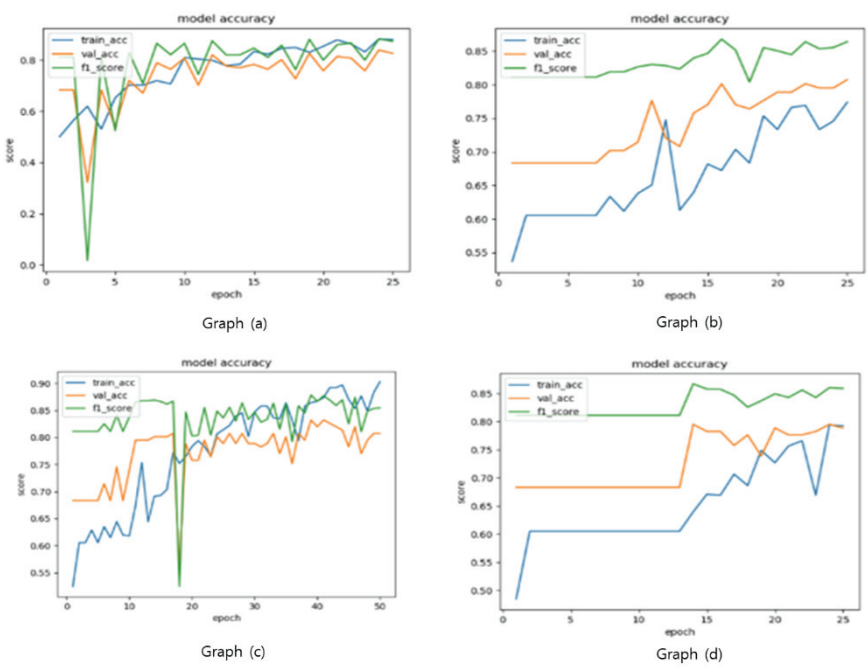

Fig. 3. CNN Model Optimization.
Fig. 3(a) dropcout:0.2, Epoch:25
Fig. 3(b) dropcout:0.6, Epoch:25
Fig. 3(c) dropcout:0.2, Epoch:50
Fig. 3(d) dropcout:0.4, Epoch:25

4.2 대상자 인식 결과

Table 2는 대상자 인식 모델의 테스트를 위해 실제 채팅방에서 특정 대상을 지칭하며 대화하는 데이터를 100개 모아 3회 실험한 결과이다.

62%에서 72%의 정확도를 보여주었으며 True Negative가 False Positive 보다 많았다. True Negative는 대부분 Twitter 과정에서 잘못 분해되거나 사전 검색 과정에서 걸리는 경우였다. False Positive는 오타 혹은 변형된 단어에 대한 경우가 대부분이다.

Table 2. The Performance of the Sequential Deletion Algorithm

Accuracy[%]	Algorithm		
	Answer [%]	True Negative [%]	False Positive [%]
62	62	29	9
72	72	23	5
68	68	21	11

4. 결 론

본 연구는 나날이 높아지는 인터넷 이용률로 온라인으로 대화하는 사람들이 많아지면서 익명의 사용자에게 비방을 듣는 경우가 빈번해지는 문제를 해결하고자 시작했다. 본 논문은 비속어를 필터링하기 위한 비속어 탐지 알고리즘과 대상자 탐지를 위한 순차적 삭제 알고리즘을 제안한다. 기존의 Masking 기반의 비속어 필터에서 판별이 힘들었던 비속어를 본 논문의 딥러닝 기반의 모델에서 필터링 할 수 있다. 또한, 구현된 모델을 채팅 어플리케이션에 이식해서 실시간 상황에서 비속어를 찾고 그 데이터를 사용자에게 보여줄 수 있으며, 또한 비속어를 사용한 대상자를 인식하는 알고리즘을 제안한다.

참고문헌

1. Gulli, A., & Pal, S., “Deep learning with Keras”, Packt Publishing Ltd., 2017.
2. Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom, “A convolutional neural network for modelling sentences”, arXiv preprint arXiv:1404.2188, 2014.
3. Jang, Baekcheol, Inhwan Kim, and Jong Wook Kim. “Word2vec convolutional neural networks for classification of news articles and tweets”, PloS one 14.8, 2019.
4. Chung, HaeKyung and Na, Jungjo, “Development of Wine Recommendation App Using Artificial Intelligence-Based Chatbot Service”, Journal of the Semiconductor & Display Technology, Vol. 18 Issue 3, pp. 93-99, 2019.
5. Ahn, Hyochang and Lee, Yong-Hwan, “A Research of CNN-based Object Detection for Multiple Object Tracking in Image”, Journal of the Semiconductor & Display Technology, Vol. 18 Issue 3, pp. 110-114, 2019.
6. Soohyun Kim, et al., “An Offensive Words Detection Method Using the Levenshtein Distance Algorithm”, KIISE Korea Software Congress 2018, pp. 2012-2014, 2018.
7. Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov, “Enriching Word Vectors with Subword Information”, 2016.
8. Park, E. L., & Cho, S., “KoNLPy: Korean natural language processing in Python”, Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, Vol. 6, pp. 133-136 2014.

접수일: 2021년 3월 12일, 심사일: 2021년 3월 15일, 게재확정일: 2021년 3월 17일