변형된 비속어 탐지를 위한 토큰 분류

고성민 [1], 신유현 [2]
[1] 인천대학교 컴퓨터공학부 학부생
[2] 인천대학교 컴퓨터공학부 교수

sm970309@inu.ac.kr, yhshin@inu.ac.kr

Token Classification for Detecting Modified Profanity

Sung-Min Ko[1], Youhyn Shin[2]
[1]Dept. of Computer Science and Engineering, Incheon National University
[2]Dept. of Computer Science and Engineering, Incheon National University

요 약

비속어 탐지 기법으로 주로 사용되는 비속어 데이터베이스 활용 방식 혹은 문장 자체를 혐오, 비혐오로 분류하는 방식은 변형된 비속어 탐지에 어려움이 있다. 본 논문에서는 자연어 처리 태스크 중 하나인 개체명 인식 방법에서 착안하여 시퀀스 레이블링 기반의 비속어 탐지 방법을 제안한다. 한국어 악성 댓글 중 비속어 부분에 대해 레이블링 된 데이터셋을 구축하여 실험을 진행하고, 이를 통해 F1-Score 약 0.88 의 결과를 보인다.

1. 서론

현대 사회에서 디지털 커뮤니케이션이 지속적으로 확장되고 이로 인해 온라인 플랫폼에서 생성되는 텍스트 데이터 양이 급격하게 증가함에 따라 건전한 온라인 환경 조성을 위한 비속어 탐지 및 관리의 필요성이 커지고 있다. 하지만 기존의 비속어 탐지와 관련된 연구들은 사전에 구축한 비속어 데이터베이스를 활용하거나[1] 문장 자체를 혐오 혹은 비 혐오 문장으로 분류하는 방식[2]으로 비속어를 탐지하기 때문에 띄어쓰기, 특수기호 삽입, 유사 발음 표기 등 변형된 비속어 탐지에 취약하다는 단점이 있다. 본 논문에서는 비속어 탐지에 BIO 태깅 방식을 활용한 토큰 분류 모델을 사용함으로써 기존 방식의 단점을 보완하고 온라인 환경에서의 비속어 문제를 완화하고자 한다.

2. 관련 연구

2.1 Electra

Electra[3]는 Generator 가 실제 문장에서 단어를 대체하여 새로운 문장을 생성하고 Discriminator 가 생성된 대체 문장을 원래 문장과 구별하는 방식으로 학습을 진행하는 LLM 이다. 본 논문에서는 Electra 기반의 사전학습모델을 사용하여 비속어 탐지 모델을 설계하였다.

2.2 NER

NER (Named Entity Recognition)[4]은 흔히 알려진 자연어 처리 태스크 중 하나로, 문장 내에서 기관, 사람 이름, 지명 등의 개체를 분류하는 작업이다. 개체명 인식에서는 주로 BIO 태깅 방식을 사용하여 레이블링을 한다. 이는 하나의 개체를 토크나이징 했을 때 시작 토큰에는 B 태그를 붙여주고, 이를 제외한 개체의 나머지 토큰에는 I 태그를 붙여주며, 그 외의 토큰에는 O 태그를 붙여주는 방식이다. 본 논문에서는 이를 활용하여 비속어를 하나의 개체로 정의하고 해당 토큰들을 분류 해냄으로써 변형된 비속어에 대해 효과

적으로 대응하여 탐지하고자 한다.

3. 설계 및 구현

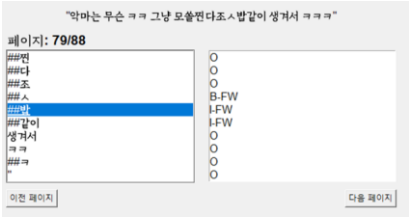
3.1 모델 설계

본 논문에서는 Electra 기반의 사전학습모델인 KcElectra[1]를 사용하여 모델을 설계하였다. KcElectra 모델은 한국어에 적용가능한 PLM 으로, 사용자 생성 및 노이즈가 있는 텍스트에 강인하다는 장점이 있다.

3.2 데이터셋

학습을 위해 GitHub 에 있는 korean-malicious-comments-dataset[2] 중 0 으로 레이블링 된 5,000 개의 문장을 사용해 데이터 셋을 구축하였다. 해당 데이터 셋을 사용한 이유는 다음과 같다. 1) 다른 데이터셋에 비해 한 문장에 비속어가 포함 되어있는 비율이 높기 때문이고, 2) 다른 데이터셋 같은 경우 비속어가 사전에 "OOO"로 필터링 되어있는 경우가 많았는데, 해당 데이터셋은 비속어 필터링이 적용되지 않은 데이터가 많았기 때문이다.

본 논문에서는 비속어 태그를 FW 로 명명하고 GUI 프로그램을 활용하여 학습을 위한 데이터셋을 구축하였다.



(그림 1) GUI 프로그램을 활용한 레이블링 예시

4. 성능 평가

다음은 데이터셋을 Train, Validation, Test 각각 8:1:1 의 비율로 나누어 학습한 결과이다. Test 데이터셋에서는 약 0.88 의 F1-Score 를 기록하였다.

[1] https://github.com/Beomi/KcELECTRA

[2] https://github.com/ZIZUN/korean-malicious-comments-dataset

Epoch	Validation Loss	Precission	Recall	F1
1	0.020738	0.882353	0.901288	0.891720
2	0.013053	0.881148	0.922747	0.901468
3	0.015659	0.883817	0.914163	0.898734
4	0.020260	0.860082	0.896996	0.878151
5	0.019786	0.875000	0.901288	0.887949

(표 1) 학습 결과

이렇게 학습한 비속어 탐지 모델을 사용하여 실제 문장에서 비속어를 추출하여 보았다.

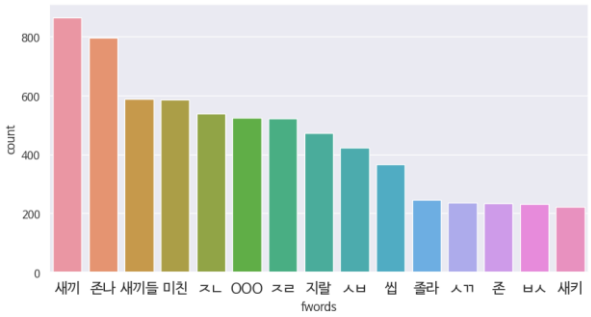
예시문장 1 - 비속어 1 개	
문장	포그바 솔직히 비호감이었는데 ㅈㄴ 매력있네 김종국도 영어 이렇게 잘하는지 첨알았음
결과	ㅈㄴ(score:0.9999)
예시문장 2 - 비속어 2 개	
문장	버논 얼굴 ㅈㄴ 잘생긴거 개씹인정... 진짜 약간 디카프리오 얼굴 느낌남
결과	ㅈㄴ(score:0.9998), ##씹(score:0.9989)

(표 2) 문장 속 비속어 추출 예시

또한 변형된 비속어에 대해서도 결과를 확인해보면 다음과 같다.

비속어 변형 예시 – 띄어쓰기	
문장	시 발ㅋㅋㅋㅋㅋㅋ
결과	시 발(score:0.9994)
비속어 변형 예시 –특수기호 사용	
문장	씨@!@#발ㅋㅋㅋㅋㅋㅋ
결과	씨@!@#발(score:0.9981)
비속어 변형 예시 –유사발음	
문장	쓰으으바 ㅋㅋㅋㅋㅋㅋ
결과	쓰으으바(score:0.9992)

(표 3) 변형된 비속어 추출 예시



(그림 2) 모델을 활용해 추출한 비속어 Count Plot

5. 결론

본 논문에서는 기존 비속어 탐지방식과 달리 토큰 분류를 활용한 방식으로 연구를 진행하였다. 모델 학습을 위해 KcElectra 를 사용하였고 korean-malicious-comments-dataset 을 사용하여 데이터셋을 구축하였으며 학습 결과 약 0.88 의 F1-Score 를 얻을 수 있었다. 학습한 모델은 실제 문장 속 비속어 추출에 좋은 성능을 보여주었는데, 이를 통해 건전한 온라인 환경 조성에 도움이 될 것으로 기대가 된다.

사사문구

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학석사연계 ICT 핵심인재양성사업의 연구결과로 수행되었음" (IITP-2023-RS-2023-00260175)

참고문헌

[1] Jongwoo Kim and Sunjeong Lee, "Developing a Connection Restrictions Filtering System for Websites based on Swear Words Extraction," Journal of KIISE, vol. 46, no. 12, pp. 1272-1278, 2019, doi: 10.5626/JOK.2019.46.12.1272

[2] Seyoung Lee and Saerom Park, "Analyzing the Classification Results for Korean Hatespeech and Bias Detection Models in Malicious Comment Dataset," Journal of the Korean Institute of Industrial Engineers, vol. 48, no. 6, pp. 636-643, 2022, doi: 10.7232/JKIIE.2022.48.6.636

[3] Clark, Kevin, et al. "Electra: Pre-training text encoders as discriminators rather than generators." *arXiv preprint arXiv:2003.10555* (2020).

[4] Li, Jing, et al. "A survey on deep learning for named entity recognition." *IEEE Transactions on Knowledge and Data Engineering* 34.1 (2020): 50-70.