The Transactions of the Korea Information Processing Society
TKIPS Vol.13, No.4, pp.181~188

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.4.181

Token-Based Classification and Dataset Construction for Detecting Modified Profanity

Sungmin Ko† · Youhyun Shin††

ABSTRACT

Traditional profanity detection methods have limitations in identifying intentionally altered profanities. This paper introduces a new method based on Named Entity Recognition, a subfield of Natural Language Processing. We developed a profanity detection technique using sequence labeling, for which we constructed a dataset by labeling some profanities in Korean malicious comments and conducted experiments. Additionally, to enhance the model's performance, we augmented the dataset by labeling parts of a Korean hate speech dataset using one of the large language models, ChatGPT, and conducted training. During this process, we confirmed that filtering the dataset created by the large language model by humans alone could improve performance. This suggests that human oversight is still necessary in the dataset augmentation process.

Keywords : Artificial Intelligence, Natural Language Processing, Named Entity Recognition, Profanity, ChatGPT

변형된 비속어 탐지를 위한 토큰 기반의 분류 및 데이터셋

고 성 민† · 신 유 현††

요 약

기존의 비속어 탐지 방법들은 의도적으로 변형된 비속어를 식별하는 데 한계가 있다. 이 논문에서는 자연어 처리의 한 분야인 개체명 인식에 기반한 새로운 방법을 소개한다. 우리는 시퀀스 레이블링을 이용한 비속어 탐지 기법을 개발하고, 이를 위해 한국어 악성 댓글 중 일부 비속어를 레이블링하여 직접 데이터셋을 구축하여 실험을 수행하였다. 또한 모델의 성능을 향상시키기 위하여 거대 언어 모델중 하나인 ChatGPT를 활용해 한국어 혐오발언 데이터셋의 일부를 레이블링을 하는 방식으로 데이터셋을 증강하여 학습을 진행하였고, 이 과정에서 거대 언어 모델이 생성한 데이터셋을 인간이 필터링 하는 것만으로도 성능을 향상시킬 수 있음을 확인하였다. 이를 통해 데이터셋 증강 과정에는 여전히 인간의 관리감독이 필요함을 제시하였다.

키워드 : 인공지능, 자연어처리, 개체명인식, 비속어, ChatGPT

1. 서 론

디지털 커뮤니케이션의 발전과 함께 온라인 플랫폼에서 생성되는 텍스트 데이터의 양이 급증하고 있다. 이에 따라 SNS 등을 통해 생성되는 비속어 데이터의 양 또한 급증하고 있는데, 비속어의 사용은 정보윤리와 자아정체성에 영향을 미치고 [1] 이는 학교폭력 등과 같은 심각한 사회 문제로도 이어질 수 있다[2]. 따라서 건강한 온라인 환경을 유지하기 위해 비속어의 탐지 및 관리가 점점 더 중요해지고 있다. 그러나 현재 대부분의 비속어 탐지 연구는 변형된 비속어를 탐지하는 데 한계가 있는데, 특히 한국어의 경우 비속어 일부를 초성으로 바꾸기, 특수 문자나 숫자 사용, 음절이나 철자 변형 등의 다양한 방법으로 변형[3]이 가능하기 때문에 비속어 탐지에 어려움을 겪는다.

본 논문은 온라인 환경에서의 비속어 문제를 효과적으로 완화하기 위해, 자연어 처리 태스크 중 하나인 개체명인식(Named Entity Recognition, NER)[4]에서 착안한 토큰 기반 분류 모델을 비속어 탐지 방법으로 제안한다. 이를 위해 시 BIO 태깅을 적용한 시퀀스 레이블링 방식으로 데이터셋을 직접 구축하였으며, ChatGPT를 활용해 데이터셋을 증강[5]하고 모델의 성능을 높이고자 하였다. 또한 그 과정에서 생성된

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학석사연계ICT 핵심 인재양성사업의 연구결과로 수행되었음(IITP-2024-RS-2023-00260175).
※ 이 논문은 2023년 ACK 2023의 우수논문으로 "변형된 비속어 탐지를 위한 토큰 분류"의 제목으로 발표된 논문을 확장한 것임.
† 비 회 원 : 인천대학교 컴퓨터공학과 석사과정
†† 정 회 원 : 인천대학교 컴퓨터공학부 부교수
Manuscript Received : December 26, 2023
First Revision : March 7, 2024
Accepted : March 22, 2024
* Corresponding Author : Youhyun Shin(yhshin@inu.ac.kr)

ChatGPT 데이터셋과 인간이 직접 생성한 데이터셋을 비교해, 데이터셋 구축과정에서 인간 감독의 필요성에 대해서도 서술하였다. 이는 기존의 방법들이 가진 한계를 개선하고 더욱 정교한 비속어 식별을 가능하게 함으로써, 건전한 디지털 커뮤니케이션 환경을 조성하는 데 기여할 것이다.

본 논문의 기여점(contribution)은 다음과 같다:

1. NER을 비속어 탐지에 적용하여 한국어 비속어에 해당하는 토큰을 탐지하는 모델을 제안하고, 이를 학습하기 위한 레이블 데이터셋을 구축하였다.

2. 거대 언어 모델(Large Language Models,LLM)을 활용하여 레이블링한 데이터셋과 사람이 직접 레이블링한 데이터셋의 차이를 살펴보고, 레이블러로써 ChatGPT의 성능에 대해 살펴보았다.

본 논문의 구성은 다음과 같다:

제 2장에서는 본 연구의 핵심 기반 기술과 함께 비속어 탐지에 관한 기존 연구들을 살펴본다. 이러한 이론적 배경을 바탕으로, 제 3장에서는 본 논문에서 제안한 모델의 구조와 학습 방법론을 소개한다. 또한, 이를 학습 시키기 위해 GUI를 활용하여 사람이 직접 구축한 데이터셋과 ChatGPT를 활용하여 증강한 데이터셋에 대해 상세히 설명한다. 제 4장에서는 본 연구에서 수행된 실험 설계 및 학습 과정을 상세히 기술하고, 실험 결과를 통해 제안된 모델의 유효성을 검증한다. 마지막으로, 제 5장에서는 연구 결과를 요약하고, 본 연구의 의의 및 향후 연구 방향에 대해 논의하며 논문을 마무리 짓는다.

2. 관련 연구

2.1 Electra

Electra[6]는 기존의 언어 모델들과 차별화되는 독특한 학습 방식을 채택한 언어 모델이다. Electra는 두 부분, 즉 Generator와 Discriminator로 구성되어 있다. Generator는 실제 문장의 일부 단어를 다른 단어로 대체하여 변형된 문장을 생성한다. 이후 Discriminator는 이 변형된 문장을 원본 문장과 구별하는 작업을 수행하며, 이 과정에서 더욱 정교한 언어 이해 능력을 학습한다. 이러한 학습 방식은 특히 문장 내의 미묘한 문맥적 차이를 파악하는 데 유리하며, 비표준적인 표현이나 변형된 어휘를 감지하는 데 효과적이다.본 연구에서는 이러한 특징을 가지는 Electra를 기반으로 하는 사전 학습 모델 (Pre-trained Language Model, PLM)을 활용하여 모델을 설계하고 실험을 진행하였다.

2.2 NER

개체명 인식(Named Entity Recognition, NER)[4]은 자연어 처리의 중요한 분야 중 하나로, 문장 내에서 기관, 인물, 지명 등 특정 개체를 식별하고 분류하는 작업이다. 이 과정에서 주로 사용되는 BIO 태깅 방식은 토크나이징된 개체명에 대해 'Beginning' (B), 'Inside' (I), 그리고 'Outside' (O)의 태그를 할당하여 개체의 경계와 유형을 구분한다. 즉, 각 개체의 시작 토큰에는 B 태그를, 개체 내의 나머지 토큰에는 I 태그를, 그리고 개체가 아닌 토큰에는 O 태그를 붙여 개체명을 명확하게 식별한다. 본 논문에서는 이러한 NER의 BIO 태깅 기법을 적용하여, 비속어를 특정 개체로 간주하고 분류하는 새로운 접근 방식을 제시한다.

2.3 LLM

거대 언어 모델(Large Language Models, LLM)은 대규모 데이터셋을 사용하여 훈련된, 고도로 발달된 자연어 처리 모델들을 말한다. 이러한 모델들은 방대한 양의 텍스트 데이터로부터 언어의 복잡한 패턴을 학습하며, 이를 통해 문장 생성, 의미 해석, 문맥 분석 등 다양한 언어적 작업을 수행할 수 있다. 최근 LLM은 그 정교함과 범용성으로 인해 다양한 자연어 처리 태스크에서 주목받고 있다. 본 논문에서는 여러 LLM 중 ChatGPT (GPT-3.5)를 활용하여 데이터셋 문장에서 비속어를 레이블링하는 방식으로 데이터 증강(Data Augmentation)[5]을 진행하였고, 이를 통해 기존 비속어 탐지 모델의 성능을 향상시키고자 하였다.

2.4 비속어 탐지 선행 연구

1) 규칙 기반 비속어 탐지

미리 정의한 비속어 데이터베이스에 의존하여, 문장 안에 해당 비속어가 존재하면 필터링을 하는 방식의 비속어 탐지 기법[7,8,9]이다. 이는 1) 비속어의 일부를 초성으로 바꾸기, 2) 특수 문자 사용, 3) 비슷한 발음으로의 치환 등의 방법으로 변형된 비속어를 탐지하는데 한계가 있다. 또한 발음과 표기는 같으나 실제 비속어가 아닌 단어를 오히려 필터링하는 한계점도 존재한다.

2) 감정 분석 기반 비속어 탐지

비속어가 포함된 문장 자체를 혐오 혹은 비혐오로 분류하여 비속어를 탐지하는 방식이다. LSTM[10], CNN[11]과 같은 딥러닝 모델과 Transformer 기반의 모델[12,13] 등을 사용하는 방식이 있으며 비속어 자체를 찾아내기가 어렵고 특수 문자 제거 등의 전처리 과정을 거쳐야 한다.

3) 이미지 기반 비속어 탐지

CNN을 기반으로 한 모델[14]로, 텍스트를 캡처한 이미지 내의 단어를 미리 정의한 금칙어 클래스로 분류하는 방식으로 비속어를 탐지하는 기법이다. 다른 모델들보다 변형된 비속어에 비교적 좋은 성능을 보였으나, 금칙어 클래스가 많아지면 모델의 성능이 낮아지고 비슷한 형태의 단어 구별에는 한계가 있다.

3. 설계 및 구현

3.1 모델 설계

본 논문에서는 Huggingface에 있는 token classification[1] 관련 코드를 baseline 코드로 사용하였고, 한국어 특화 사전 학습 모델인 KcElectra[2]를 기반으로 비속어 탐지 모델을 개발하였다.

1) KcElectra

KcElectra는 한국어에 최적화된 언어 모델로, 특히 사용자가 생성한 다양한 형태의 텍스트와 노이즈가 포함된 언어 데이터에 강한 내성을 가진다는 특징을 지닌다. 이 모델은 한국어의 복잡한 문법 구조와 다양한 표현을 효과적으로 처리할 수 있으며, 이를 통해 비표준적이거나 변형된 비속어를 식별하는 데 있어 뛰어난 성능을 발휘한다. 따라서, KcElectra를 활용한 비속어 탐지 모델은 한국어 텍스트의 특성을 깊이 이해하고, 정교한 비속어 식별이 가능하다는 중요한 장점을 가진다.

2) Token alignment

먼저 비속어 태그는 FW로 명명하였다. 또한 BIO 방식을 사용하여 비속어의 시작 토큰은 B-FW, 비속어 내의 나머지 토큰은 I-FW, 그 외에 토큰은 O 태그로 레이블링 하였다. 모델을 학습시키기 위해서는 특정 토큰을 특정 인덱스로 매핑해 줘야한다. 이에 [CLS], [SEP] 와 같은 스페셜 토큰은 -100으로, O 태그는 0, B-FW 태그는 1, I-FW 태그는 2로 매핑하였다.

3) Hyperparameter

모델 학습에 사용한 하이퍼파라미터는 다음과 같다.

Table 1. Token-based Labeling Example

문장	아 이 개병신새끼씨발ㅋㅋ
Tokenizing	['아', '이', '개', '##병', '##신', '##새끼', '##씨', '##발', '##ㅋㅋ']
Label	('O', 'O', 'B-FW', 'I-FW', 'I-FW', 'I-FW', 'B-FW', 'I-FW', 'O')

Table 2. Hyperparameter

Learning rate	1e-04
Batch Size	32
Epoch	10
Adam epsilon	1e-08
Warmup ratio	0.1
Weight Decay	0.01

3.2 데이터셋 구축

1) 한국어 악성 댓글 데이터셋

먼저 비속어 탐지 모델 학습을 위해 GitHub에 있는 한국어 악성 댓글 데이터셋(korean-malicious-comments-dataset[3])을 사용하여 데이터셋을 구축하였다. 데이터셋 중 0(negative)으로 레이블링 된 5,000개의 문장을 토크나이징 한 후에 있는 비속어가 포함된 토큰을 찾아 레이블링하였고, 이를 위해 GUI 프로그램을 활용하였다.

데이터셋의 품질 향상을 위해 명확한 기준을 정해놓고 레이블링을 진행하였는데, 그 기준은 다음과 같다.

a) 비속어 자체만 레이블링하고, 그 외 혐오를 담고 있지만 욕이 아닌 단어는 레이블링 하지 않는다.
예: OO충, 한남.녀, 돼지 등

b) 접두사로 붙는 비속어는 접두사만 레이블링 한다.
예: '존'잘, '씹'노잼 등

c) 다른 단어와 결합된 비속어는 레이블링하지 않는다.
예: 엠빙신(MBC를 의미), 개독(기독교를 의미) 등

2) 한국어 혐오 발언 데이터셋

다음으로 ChatGPT를 활용하여 레이블링과 변형된 비속어를 생성하기 위해 Hugging Face에 있는 한국어 혐오 발언 데이터셋(jeanlee/kmhas_korean_hate_speech[4])을 사용하였다. 79,000개의 학습 데이터셋 중 비속어가 포함된 문장을 추출하였고 총, 약 5,000개의 문장을 추출할 수 있었다. ChatGPT를 활용한 레이블링을 위해 다음과 같은 프롬프트를 사용하였다.

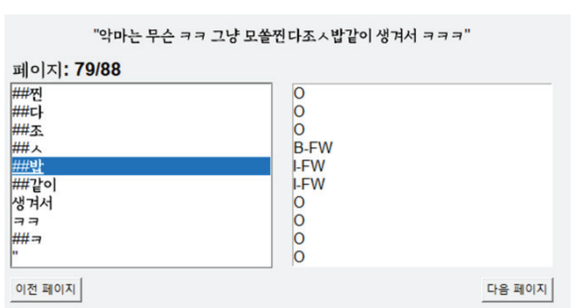


Fig. 1. GUI Program

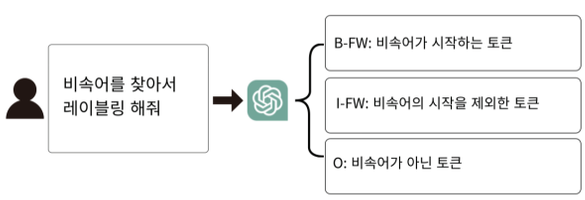


Fig. 2. Labeling Prompt

1) huggingface.co/docs/transformers/tasks/token_classification
2) github.com/Beomi/KcELECTRA
3) github.com/ZIZUN/korean-malicious-comments-dataset
4) huggingface.co/datasets/jeanlee/kmhas_korean_hate_speech

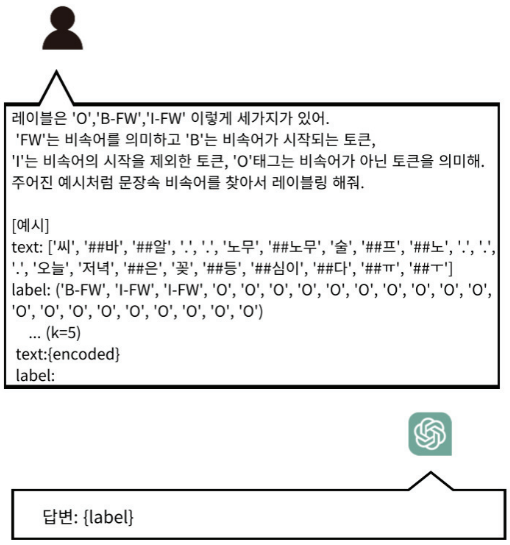


Fig. 3. Prompt Example (k=5)

Table 3. Example of Labeling Using ChatGPT

문장	찌질한 새끼 인정 씹ll년도 인정
Tokenizing	['찌질한', '새끼', '인정', '씹', '##ll', '##년도', '인정']
Label	('O', 'B-FW', 'O', 'B-FW', 'I-FW', 'I-FW', 'O')

Table 4. Example of Incorrect Labeling

출력 길이 제한으로 레이블링이 끝까지 되지 않은 경우	
Tokenizing	['전국민이', ',', '언론이', ',', '정치인이', ',', '사회가', '지지해', '##주는데', ',', '지지해', '##주는', '사람이', '없어서', '외', '##로웠', '##다고', '?', '말이', '되는', '소리를', '해라', '.', '그리고', '너가', '자리', '##한', '것도', '아니고', '.', '.', '.', '그랬다', '##더라', ',', '그렇게', '들었다', '.', '.', '.', '예혀', '.', '.', '미투', '##가', '아니라', '미친', '여인', '##들의', '투기', '##장이', '##다', '.']
Label	('O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
토크나이징 된 단어와 레이블의 길이가 같지 않은 경우	
Tokenizing	[' ', '미친', '무뇌', '##가', '설치는', '개', '##ㅂ', '##ㅅ', '나라', ' '] - 길이 10
Label	('O', 'B-FW', 'I-FW', 'O', 'O', 'O', 'B-FW', 'I-FW', 'O') -길이 9

또한 일관된 답변의 형식과 답변의 질을 향상시키기 위해 적절한 예시가 필요하였다. 따라서 k=5인 few-shot learning [15]을 진행하였다.

제시한 프롬프트를 사용하여 문장을 레이블링 하였고, 그 예시는 다음과 같다.

하지만 잘못 레이블링된 데이터셋이 있었고, 이를 제외한 총 3,000개의 데이터셋을 사용하였다.

4. 학습 및 평가

4.1 실험 설계

실험에 사용한 데이터셋의 표기법은 다음과 같다.

사람이 레이블링한 데이터셋을 사용하는 경우 Human으로, ChatGPT가 레이블링한 데이터셋을 사용하는 경우 GPT로, 둘을 함께 사용하는 경우 HumanGPT로 표기하였다. 또한 ChatGPT를 이용해 레이블링 후 인간이 필터링한 데이터셋을 사용한 경우 중간에 f를 추가하여 표기하였고, 사용한 학습 데이터의 개수를 함께 표기하였다.

예시) HumanGPT_f_4300: Human이 GUI 프로그램을 활용하여 직접 레이블링한 데이터에, ChatGPT를 이용해 레이블링 후 필터링한 데이터셋으로 증강시킨 데이터셋이며 학습 데이터 개수는 총 4,300(4,000+300)개이다.

1) Human Labeling

먼저 사람이 직접 레이블링한 데이터셋만을 사용하여 비속어 탐지 모델 학습을 진행하였다. 3.2.1에서 구축한 데이터셋을 각각 8:1:1의 비율로 train(4,000), validation(500), test(500) set으로 나누어 사용하였으며, ChatGPT 데이터셋과의 비교를 위하여 3000, 300개의 train set으로도 실험을 진행하였다. validation, test set은 이후 실험에서도 동일하게 사용하였다.

2) ChatGPT Labeling

다음으로 ChatGPT를 활용해 레이블링한 데이터셋을 사용하여 비속어 탐지 모델 학습을 진행하였다. 3.2.2에서 구축한 3,000개의 데이터셋(ChatGPT_3000)과, 그 중 사람이 직접 필터링을 한 300개의 데이터셋(ChatGPT_f_300)을 train set으로 사용하였다. 필터링을 진행한 기준은 3.2.2의 a), b), c) 이다. 공정한 비교를 위해 validation과 test set은 Table 5와 동일한 데이터를 이용하였다.

Table 5. Statistics on Human Labeled Dataset

dataset	train	validation	test
Human_4000	4,000	500	500
Human_3000	3,000	500	500
Human_300	300	500	500

Table 6. Statistics on ChatGPT Labeled Dataset

dataset	train	validation	test
GPT_3000	3,000	500	500
GPT_f_300	300	500	500

3) Human+ChatGPT Labeling

다음으로 ChatGPT로 증강한 데이터셋을 사용하여 비속어 탐지 모델을 학습하였다. 인간이 필터링한 ChatGPT 데이터셋 300개(ChatGPT_f_300)와 3,000개의 ChatGPT 데이터셋(ChatGPT_3000) 중 무작위로 샘플링한 데이터셋 300개(ChatGPT_300)를 사용했으며, 각각의 경우에 대해 100개 단위로 나누어 실험을 진행하였다.

마지막으로 증강된 데이터가 많을수록 모델의 성능이 함께 증가하는지 확인하기 위해 필터링하지 않은 3,000개의 ChatGPT 데이터셋을 500개씩 늘려가며 비속어 탐지 모델 학습을 진행하였다.

실험에서 사용한 모델의 평가지표로 F1 score를 사용하였다. F1 score는 정밀도(Precision)와 재현율(Recall)의 조화 평균을 나타내는 평가지표로, 데이터셋의 불균형이 있을 때 특히 유용한 지표이다. 본 논문에서 사용하는 데이터셋의 경우 비속어 태그인 B-FW, I-FW 태그에 비해 O태그의 비율이 훨씬 높은 불균형 데이터셋이기 때문에 모델의 평가지표로 F1 score를 사용하였다.

Table 7. Statistics on Human Labeled Dataset with Data Augmentation(ChatGPT Filtered Dataset)

dataset	train	validation	test
HumanGPT_f_4100	4,000+100	500	500
HumanGPT_f_4200	4,000+200	500	500
HumanGPT_f_4300	4,000+300	500	500

Table 8. Statistics on Human Labeled Dataset with Data Augmentation(ChatGPT Random Sampled Dataset)

dataset	train	validation	test
HumanGPT_4100	4,000+100	500	500
HumanGPT_4200	4,000+200	500	500
HumanGPT_4300	4,000+300	500	500

Table 9. Statistics on Human Labeled Dataset with Data Augmentation(ChatGPT All Dataset)

dataset	train	validation	test
HumanGPT_4500	4,000+500	500	500
HumanGPT_5000	4,000+1,000	500	500
HumanGPT_5500	4,000+1,500	500	500
HumanGPT_6000	4,000+2,000	500	500
HumanGPT_6500	4,000+2,500	500	500
HumanGPT_7000	4,000+3,000	500	500

$$\mathrm{Precision} = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Positive(FP)} \quad (1)$$

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)} \quad (2)$$

$$F1score = 2 \times \frac{\mathrm{Precision} \times Recall}{\mathrm{Precision} + Recall} \quad (3)$$

4.2 실험 결과

실험은 각 모델에 대하여 10번씩 진행하였으며 모델에 대한 평균 F1 score를 사용하여 성능을 비교하였다.

1) Human Labeling

인간이 레이블링한 데이터셋의 경우 데이터셋의 크기에 비례한 F1 score를 얻을 수 있었다.

2) ChatGPT Labeling

ChatGPT가 레이블링한 데이터셋의 경우 F1 score가 데이터셋의 크기에 비례하지 않고, 오히려 더 적은 양의 데이터셋으로 학습한 모델에서 더 좋은 결과를 보였다.

이때 해당 모델(ChatGPT_f_300)은 Fig. 4와 같이 오히려 인간이 레이블링한 동일한 크기의 데이터셋보다 좋은 성능을 보였다. 이는 데이터셋 증강 과정에서 인간의 감독과 정제과정이 중요함을 나타낸다.

Table 10. Experimental Result on Human Labeled Dataset

dataset	train	F1 score
Human_4000	**4,000**	**0.8693**
Human_3000	3,000	0.7962
Human_300	300	0.5346

Table 11. Experimental Result on ChatGPT Labeled Dataset

dataset	train	F1 score
GPT_3000	3,000	0.1826
GPT_f_300	**300**	**0.5893**

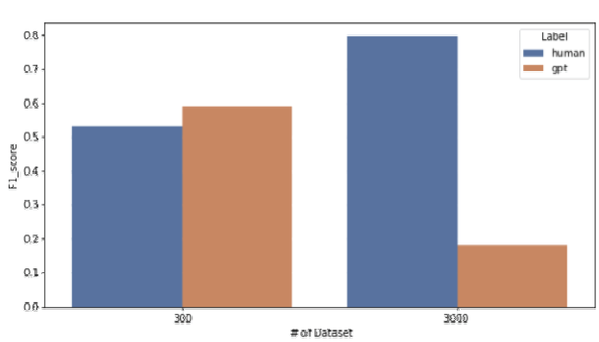


Fig. 4. Compares the F1 Score with Different Numbers of Human and ChatGPT Dataset

3) Human+ChatGPT Labeling

인간이 레이블링한 데이터셋과 ChatGPT가 레이블링한 데이터셋을 함께 사용하는 경우 필터링한 데이터셋의 경우 기존 모델(human_4000)과 성능이 비슷하거나 더 향상된 결과를 보였으며 특히 humangpt_f_4100의 경우 평균 0.89의 F1 score를 기록하였다. 그러나 필터링을 하지 않은 데이터셋은 성능이 기존 모델보다 떨어지는 결과를 보였다.

이와 같은 결과가 나오는 이유는 모델을 학습시킬 때 필터링되지 않은 데이터셋 중 낮은 품질의 데이터가 포함되어있을 경우 노이즈로 작용하여 학습 성능을 떨어뜨리기 때문이다. 실제로 각 모델의 성능을 시각화하면 필터링을 하지 않은 데이터셋으로 학습시키는 경우 성능이 급격하게 안 좋아지는 경우를 확인할 수 있다.

또한 필터링하지 않은 ChatGPT 데이터셋(GPT_3000)을 500개씩 추가해가며 학습시킨 결과 데이터셋의 개수가 증가할수록 성능이 하락하는 결과를 보였다. 이는 마찬가지로 데이터셋이 추가될수록 필터링되지 않은 데이터셋의 비율이 증가하여 그에 따라 포함된 낮은 품질의 데이터가 노이즈로 작용하기 때문이다.

Table 12. Experimental result on Human and ChatGPT Constructed Dataset

dataset	train	F1 score
HumanGPT_ f_4100	4,000+100	**0.8938**
HumanGPT_f_4200	4,000+200	0.8719
HumanGPT_f_4300	4,000+300	0.8743
HumanGPT_4100	4,000+100	0.7758
HumanGPT_4200	4,000+200	0.8374
HumanGPT_4300	4,000+300	0.8239

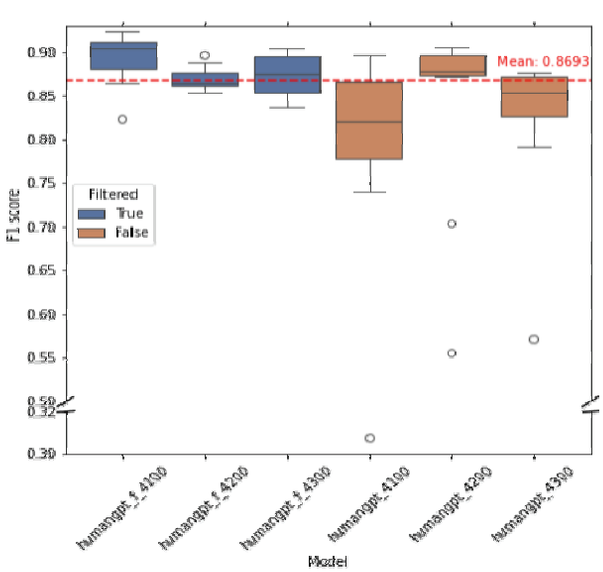


Fig. 5. Compares the HumanGPT F1 Score According to the Presence or Absence of Filtering

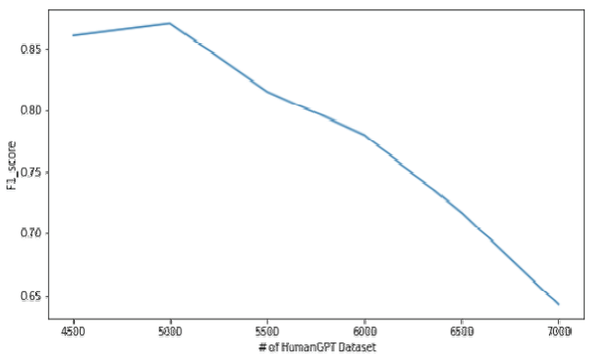


Fig. 6. HumanGPT F1 Score Relative to the Number of Dataset without Filtering

4.3 비속어 탐지

가장 높은 F1 score를 기록한 HumanGPT_f_4100 모델을 사용하여 여러 예시 문장 속 비속어를 추출해 보았고, 그 결과는 다음과 같다.

예시문장 1을 통해 맥락상 '존나'라는 비속어의 일부를 'ㅇ'으로 의도적으로 변형시킴으로써 비속어 탐지를 회피하는 경우에 대하여 비속어를 적절히 탐지함을 확인할 수 있다.

예시문장 2를 통해 비속어 중간에 @,!,# 등의 특수기호 등을 넣음으로써 비속어 탐지를 회피하는 경우에 대하여 비속어를 적절히 탐지함을 확인할 수 있다.

Table 13. Example of Extracting Profanity from a Sentence

예시문장 1 - 비속어의 일부를 변형시키는 경우	
문장	누군가했더니 고등랩퍼에서 랩 존ㅇ 못하는데 지가 최고라고 자뻑하던 애네 ㅋㅋ
결과	존ㅇ(score:0.8533)
예시문장 2 - 비속어 중간에 특수기호등을 넣는 경우	
문장	시@!#!발 드라마에서 같은 교도소라면 거기가 좋은 곳으로 받아 주는 사람들 있을거 같다 진짜 교도소가 그렇냐
결과	시@!#!발(score:0.7371)
예시문장 3 - 비속어를 비슷한 발음으로 바꾸는 경우	
문장	익명이라고 진짜 아무 글이나 싸지르는 색희들 조올라 싫다 인간들ㅇ1 다 왜 이모양이냐? 지 화나는 걸 왜 다른 사람한테 푸냐? 왜 남의 댓글 복사해서 미친짓을 하지 ㅡ. ㅡ 싸이콘가
결과	색희(score:0.9208) ,조올라(score:0.9302), 미친짓(score:0.96489)
예시문장 4 - 비속어를 숫자 등으로 대체하는 경우	
문장1	적당히 해라 18 진짜로
결과	18(score:0.8536)
문장2	올해 지방공무원 만6천여 명 채용…18살도 7급 응시 가능
결과	비속어 태그 없음

예시문장 3을 통해 '새끼'를 '색희', '졸라'를 '조올라' 등, 음절이나 철자를 의도적으로 변형시킴으로써 비속어 탐지를 회피하는 경우에 대하여 비속어를 적절히 탐지함을 확인할 수 있다.

예시문장 4-1의 경우 '씨발' 이라는 비속어를 '18'이라는 숫자로 변형시킴으로써 비속어 탐지를 회피하는 것을 적절히 탐지한데 반해, 예시문장 4-2의 경우 같은 '18'이지만 비속어로 인식하지 않는 것을 확인할 수 있다.

5. 결 론

본 논문에서는 전통적인 비속어 탐지 방식과 다르게 NER 기법을 적용해 비속어를 탐지하는 방식으로 연구를 진행하였다. 이를 위해 KcElectra 기반의 사전 학습 모델을 사용하였고, 한국어 악성 댓글 데이터셋을 활용하여 데이터셋[5]을 직접 구축하였다. 이는 기존에 없던 비속어만을 레이블링한 데이터셋이기 때문에 추후 비속어 탐지 연구에 활용될 것으로 기대된다. 또한 한국어 혐오 발언 데이터셋과 LLM을 활용하여 데이터 증강기법을 통해 모델의 성능을 향상시키고자 하였으며, 실제 약 0.89의 향상된 F1 score를 얻을 수 있었다. 이 과정에서 LLM이 생성한 데이터셋을 인간이 필터링하여 사용하는 것만으로도 성능을 향상시킬 수 있음을 확인하였다. 이를 통해 데이터셋을 증강함에 있어 중요한 것은 데이터셋의 양보다는 데이터셋의 품질임을 확인할 수 있었고, 데이터 증강 과정에서 인간 감독의 필요성을 제시하였다. 이 연구를 통해 건전한 디지털 커뮤니케이션 환경을 조성하는 데 도움이 될 것으로 기대가 된다.

References

[1] H.-G. Kim and H.-K. Kim, "An empirical study of SNS Language Violence(Expletives, Slang)," *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities and Sociology*, Vol.9, No.3, pp.99-108, 2019.

[2] H.-Y. Kang, "The mediating effect of self-control on the relationship between adolescents' usage of slang and aggression," *Journal of the Korea Institute of Youth Facility and Environment*, Vol.19, No.4, pp.33-42, 2021.

[3] D. Kim, J. Kim, and E. Kwak, "A study on the translation strategies of swearing in the game: Focusing on the semantic · phonological/morphological variations of game forbidden words," *The Journal of Interpretation and Translation Education*, Vol.21, No.3, pp.5-25, 2023.

[4] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition: Extended abstract," *IEEE 39th International Conference on Data Engineering (ICDE)*, pp.3817-3818, 2023.

[5] H. Dai et al., "Chataug: Leveraging chatgpt for text data augmentation," *arXiv preprint arXiv:2302.13007*, 2023.

[6] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[7] A. Chaudhari, P. Davda, M. Dand and S. Dholay, "Profanity detection and removal in videos using machine learning," *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pp.572-576, 2021.

[8] J. Kim and S. 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.

[9] Y.-L. Choi, J.-W. Kim, and J. Han, "Development of profanity response module for artificial intelligence service for english education," *Journal of Korean Institute of Intelligent Systems*, Vol.31, No.3, pp.192-197, 2021.

[10] M. Yi, M. Lim, H. Ko, and J. Shin, "Method of profanity detection using word embedding and LSTM," *Mobile Information Systems*, Vol.2021, 2021.

[11] Y. Kim, H. Gang, S. Han, and H. Jeong, "Swear word detection through convolutional neural network," *Proceedings of the Korea Information Processing Society Conference*, Vol.28, No.2, pp.685-686, 2021.

[12] S. Lee and S. 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.

[13] J. H. Choi, "Design and implementation of abuse sentence detecting system based on deep learning," Master's dissertation, Hanyang University, Korea, 2020.

[14] J. Yoo, "A study on the improvement of text filtering based on image learning," Master's dissertation, Sungkyunkwan University, Korea, 2019.

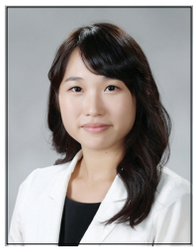
[15] T. Brown et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, Vol.33, pp.1877-1901, 2020.

[16] S. Ko and Y. Shin, "Token Classification for Detecting Modified Profanity," *Proceedings of the Annual Conference of Korea Information Processing Society Conference (KIPS) 2023*, Vol.30, No.2, pp.498-499, 2023.

5) https://github.com/sminu24/Detecting_Modified_Profanity

고 성 민

https://orcid.org/0009-0002-1303-1319
e-mail : sm970309@inu.ac.kr
2023년 인천대학교 컴퓨터공학과(학사)
2024년 ~ 현 재 인천대학교 컴퓨터공학과 석사과정
관심분야 : 자연어 처리, 인공지능



신 유 현

https://orcid.org/0000-0001-7013-9057
e-mail : yhshin@inu.ac.kr
2019년 서울대학교 컴퓨터공학부(박사)
2020년 ~ 현 재 인천대학교 컴퓨터공학부 부교수
관심분야 : 자연어 처리, 음성 언어 이해, 인공지능