Regular paper | Journal of the KIECS. pp. 1147-1152, vol. 19, no. 6, Dec. 31. 2024, t. 128, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2024.19.6.1147

KoBERT 기반 비속어 검출 모델 및 FAST API 서버 구현

김영민* · 박승민**

Implementation of KoBERT-based profanity detection model and FAST API server

Young-Min Kim* · Seung-Min Park**

요 약

본 논문에서는 한국어 BERT(KoBERT)를 전이 학습하여 비속어가 포함된 문장과 그렇지 않은 문장을 구별하는 모델을 구축하고, 이를 Python의 FAST API를 이용하여 웹 서비스 형태로 구현한 연구 결과를 제시한다. 데이터 셋은 다양한 온라인 커뮤니티와 소셜 미디어에서 수집한 문장을 활용하였으며, 전처리 과정을 거쳐 비속어 여부로 라벨링 하였다. KoBERT를 기반으로 한 분류 모델을 구축하고, 전이 학습 기법을 통해 높은 정확도의 비속어 검출 성능을 달성하였다. 또한, FAST API를 이용하여 클라이언트로부터 POST 요청을 받아 텍스트 데이터를 처리하고, 비속어 여부를 반환하는 웹 서비스를 구현하였다. 본 연구는 KoBERT를 활용한 비속어 검출의 가능성을 확인하고, 실용적인 웹 서비스 구현을 통해 실제 적용 가능성을 제시하였다. 향후 연구로는 더 다양한 데이터 셋을 활용한 모델 성능 개선과 실시간 비속어 필터링 시스템 구현을 목표로 한다.

ABSTRACT

This paper presents a study in which a model is built to distinguish between sentences containing profanity and those that do not, by applying transfer learning to KoBERT (Korean BERT). The model is implemented as a web service using Python's FAST API. The dataset consists of sentences collected from various online communities and social media platforms, and after a preprocessing stage, the sentences were labeled based on the presence of profanity. A classification model was built using KoBERT, and by utilizing transfer learning techniques, high accuracy in profanity detection was achieved. Additionally, a web service was implemented using FAST API, which processes text data received through POST requests from clients and returns whether profanity is present or not. This study confirms the potential of using KoBERT for profanity detection and demonstrates the feasibility of practical application through the implementation of a web service. Future research will aim to improve model performance by utilizing more diverse datasets and to implement a real-time profanity filtering system.

Keyword

Chain of Thought, Classification System, Korean BERT, Large Multimodal Model, Profanity Classification
CoT, 비속어 분류, koBERT, LMM, 분류 시스템

* 동서대학교 연구원(sminpark@dongseo.ac.kr)
** 교신저자 : 동서대학교 소프트웨어학과
· 접 수 일 : 2024. 10. 08
· 수정완료일 : 2024. 11. 08
· 게재확정일 : 2024. 12. 12
· Received : Oct. 08, 2024, Revised : Nov. 08, 2024, Accepted : Dec. 12, 2024
· Corresponding Author : Seung-Min Park
Dept, Dongseo University,
Email : sminpark@dongseo.ac.kr

I. 서 론

최근까지 증가하고 있는 사이버 모욕 범죄에 대한 대안이다. 사람들이 모이는 SNS는 사이버 모욕 범죄를 제제하기 위해 많은 관리 인력과 자본을 사용하고 있다. 특히 이러한 사이버 범죄는 코로나를 기점으로 4년 간 약 73% 증가 하였다. 미디어 매체에 많이 노출되는 연예인들이 기존의 사이버 모욕 범죄의 표적이였다면 최근 일반인을 대상으로 하는 '나는 솔로'와 같은 프로그램이 많아지면서 연예인뿐 만 아니라 일반인도 사이버 모욕 범죄에 노출되고 있다.

본 논문이 초점을 맞춘 것은 사이버 범죄 중 모욕 범죄와 통신매체음란법이다. 기존의 모욕죄는 공연성, 특정성, 모욕성 등 3가지 조건이 만족되어야 한다[1-2]. 누군가를 특정하고 모욕적이 언어를 사용하면 모욕 범죄가 성립된다. 본 논문은 모욕성이 성립하지 못하도록 koBERT 모델을 이용하여 문장을 Tokenizing하여 각 구문별로 단어를 분리한다. 분리된 단어에서 모욕적으로 인식될 수 있는 단어가 있을 경우 이를 Classifier로 분류하여 Clean, Curse, Conflict Of Generation, Insult, Caricature 등 5개의 class 중 하나로 분류 한다[3]. 서버는 분류된 class는 FAST API를 이용하여 요청한 클라이언트에게 반환하여 API를 개발 하였다. koBERT는 문장을 분리하여 토큰으로 나누는 LLM 기반의 모델이다. 본 논문에서는 koBERT 모델을 fine-tuning을 진행한다. fine-tuning에 사용된 데이터는 SNS, 커뮤니티에서 작성된 댓글로 익명성이 보장되는 사이트와 익명성이 보장되지 않는 사이트에서 채취하였다. 이는 편향되지 않은 데이터를 만들기 위한 방법 중 하나로 비속어가 포함되어 글을 읽는 사람으로 하여금 비관적인 글과 비속어가 전혀 포함되지 않는 글을 모두 채취하였다. 본 논문에서는 인터넷 사용자들이 쉽게 노출될 수 있는 비속어에 노출되는 것을 줄이고 SNS를 유지보수하는 개발자의 입장을 고려하여, 사용이 편리한 API 서버를 구축하게 되었다.

본 논문에서는 웹 사이트 사용자가 남긴 댓글을 기반으로 전처리 한 후 5가지의 분류 데이터 셋을 만든다[11]. 데이터 셋을 KoBert 모델에 fine-tuning하여 생성형 모델을 만들고 python의 fast API 라이브러리를 이용하여 수신 받은 데이터를 분류한 값을 수신측에 송신한다.

II. 선행연구

2.1 A Swearword Filter System for Online Game Chatting

선행 연구로서 본 논문에서는 온라인 게임에서 발생하는 언어 폭력 문제를 해결하기 위해, 자동으로 비속어를 감지하고 차단하는 시스템을 제안한 연구를 다루고 있다. 연구자는 온라인 게임의 채팅 데이터를 수집하여 비속어가 포함된 문장과 정상 문장으로 수동 분류한 뒤, 음절 n-gram과 어휘-품사 쌍을 자질로 활용했다. 카이제곱 통계량을 통해 중요한 자질을 선택하고, 이를 바탕으로 지지벡터기계(SVM)를 학습시켜 각 문장의 비속어 포함 여부를 분류하였다. 실험 결과, 약 90.4%의 F1 정확률을 기록하며 높은 성능을 입증하였다. 이 연구는 온라인 게임 내 언어 폭력 문제를 기술적으로 해결하기 위한 기초 자료로 활용될 수 있다[4].

III. 설 계

3.1 개발환경

본 논문에서 연구 진행에 사용된 언어 및 라이브러리의 환경과 버전은 Python(3.10.12), Mxnet-mkl(1.6.0), Numpy(1.23.1), Gluonnlp(0.10.0), Pandas(2.0.3), Tadm(4.66.4), Sentencepiece(0.1.99), Transformers(3.0.2), Torch(1.0.0), FastApi(0.0.4), Uvicorn(0.30.1), Nest_Asyncio(3.7.1), KoBert(0.2.3) 등이 사용되었다. 연구 환경을 Google의 Colab을 이용하여 fine-tuning을 진행했고, 5000개의 댓글 데이터를 이용하여 모델 학습을 진행했다.

3.2 주요 기능 '

3.2.1 텍스트 데이터 추출 과정 및 라벨링

본 논문의 데이터 추출 과정은 악질 커뮤니티 댓글 데이터 셋과 SNS의 댓글을 섞어서 사용하였다. 악질 커뮤니티 댓글 데이터 셋은 GitHub에 등록된 한국어 데이터로 여러 커뮤니티에 악질 댓글을 모아 놓은 데이터 셋이다.

표 1. 데이터 분류 기준

Table 1. Data Classification Criteria

Classifier	standard
Clean	If it is not included in the four categories below and the negative word among the words does not refer to the target
Curse	Contains direct profanity words
Conflict Of Generation	If it contains words that degrade each generation, age group, etc., or words that support or degrade each generation.
Insult	If a new word or derogatory word is judged as an abusive word depending on its severity after reading the sentence.
Caricature	In cases where a specific target can be designated and the target can be made into a laughing stock (mock) by others.

입력 받은 데이터가 비속어가 포함되지 않은 것은 학습하여 모델의 성능을 향상하기 위해 비속어가 포함되지 않은 댓글을 SNS에서 수집하여 추가로 데이터를 보안하였다[5]. 라벨링 과정에서는 서론의 내용처럼 문장을 5가지로 분류한다. Clean, Curse, Conflict Of Generation, Insult, Caricature 등 5가지의 분류 기준은 각각 표 1과 같다.

표 1의 기준대로 분류하여 각 데이터 마다 라벨링을 진행한다. 그 후 라벨링이 완료된 1차 가공된 데이터 셋은 사슬 추론 기법(CoT)를 이용하여 2차 가공을 진행한다[6-7].

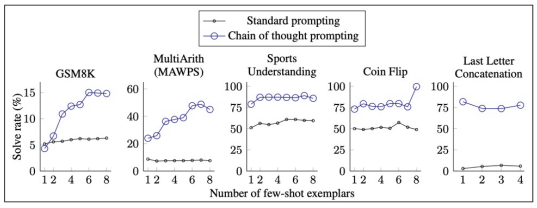

그림 1. 사슬 추론 기법(CoT) 성능 평가도

Fig. 1 Performance evaluation of the Chain of Thought (CoT) technique

그림 1은 인공지능(AI) 모델 성능 평가 지표다. 지표를 보면 Standerd prompting을 CoT Prompting로 변경했을 경우 유의미한 성능 향상이 이루어졌다. 사슬 추론 기법(CoT)의 CoT Prompting은 Prompt에 질문과 답을 사슬처럼 연결하는 기법으로 LLM의 문장 해석력과 산수 능력을 향상하는 기법이다. 본 논문에서는 사슬 추론 기법(CoT)를 LLM의 분류 시스템에 적용하여 정확도를 높인 것이다.

각 데이터 마다 Prompt-> 해석 -> Prompt 사슬 같이 연결된 데이터로 2차 가공을 진행하여 fine-tuning에 필요한 전처리 데이터를 가공하였다.

Ⅳ. 구 현

4.1 KoBert Model fine-tuning

전처리 과정을 마친 데이터를 KoBert 모델에 학습시키기 위해선 Tokenizer되기 전 5가지의 분류를 각각 0~4번까지 번호를 적용하여 KoBert가 fine-tuning을 진행할 수 있는 새로운 데이터로 만든다[8]. KoBert가 fine-tuning을 진행할 수 있도록 Max_len = 64, Batch_size = 64, Warmup_ratio = 0.1로 설정하고 KoBert Model 내부의 Tokenizer의 Tokenize를 NLP의 Vocab과 함께 적용하여 Data_train을 생성했다. Torch DataLoader 함수에 Data_train, batch_size를 파라미터로 Train_dataloader를 생성한다. 이후 AdamW함수에 파라미터를 model의 named_parameters함수가 Tokenizer 배열을 순회 하도록 하여 Optimizer Grouped Parmeters를 생성하고 lr은 앞서 Learning_rate 변수로 하여 AdamW함수를 실행해 Optimizer를 생성한다.

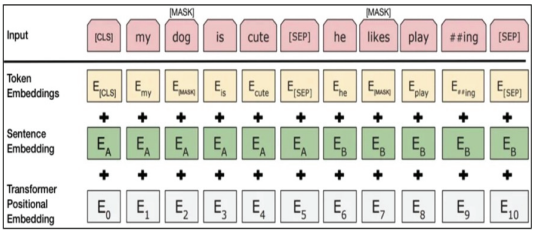

그림 2. LLM 문법 이해 방법

Fig. 2 LLM Grammar Understanding Method

그림 2는 문장을 Tokenizer하여 단어 별로 Tokenizer하여 문장을 이해하는 LLM을 시각화한 것이다. 문장을 Tokenizer하여 배열로 만들고 순회하여 각 방의 텍스트를 백터화하여 노드의 가중치를 조절한다[9-10].

약 5000개의 데이터 학습하며 가중치를 조절한 모델은 입력된 문장을 5가지의 케이스로 반환하는 모델로 fine-tuning 진행 후 모델을 Colab 드라이버에 저장한다. 주피터 노트북의 특성상 모델을 저장하지 않으면 백엔드 연결이 끊어지면서 fine-tuning된 모델의 가중치가 삭제된다.

4.2 KoBert Model 기반 API

KoBert Model fine-tuning에서 제작된 모델을 기반으로 API를 제작한다. Google Colab을 이용해서 파이썬 API 라이브러리인 FastAPI로 API서버를 구축한다. Colab은 주피터 노트북 환경이므로 Localhost로 접속할 수 없다. 따라서 FastAPI의 공용 URL로 테스트 환경을 구현한다. 실험 환경의 CORS는 Localhost:3000 포트로 설정해 CORS 위반하는 경우를 제한한다.

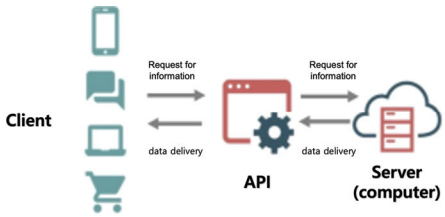

그림 3. 클라이언트&서버 통신
Fig. 3 Client & Server Communication

그림 3은 클라이언트-서버 통신 구조를 시각화 한 것으로 클라이언트가 서버 측으로 요청하면 Middleware를 거쳐 백엔드 서버로 파라미터가 전달된다. 2.2.2 KoBert Model fine-tuning 과정에서 저장된 모델을 API 서버로 호출하고 Router를 POST로 설정하여 BODY에 {Text: “문자열”} JSON 구조로 송신받는다. 설정된 JSON 형태가 아니라면 예외로 처리한다. 올바른 JSON을 수신 받을 경우 POST BODY에서 Text를 추출하여 Tokenizer을 진행한다. Tokenizer된 Text를 학습된 모델의 파라미터로 설정한다. 학습된 모델의 가중치는 Text Tokenizer를 순회하며 가중치를 연산한 후 새로운 변수에 값을 반환한다. POST Return은 {Text: “원본 문장”, classification: “반환된 변수”} JSON 구조로 최종 반환하면 POST 통신이 종료된다.

Ⅴ. 검 증

5. 시뮬레이션

5.1 모델 성능 평가

본 논문은 5000개의 데이터로 fine-tuning을 진행했고, 약 100개의 테스트 데이터로 모델 성능 평가를 진행하였다.

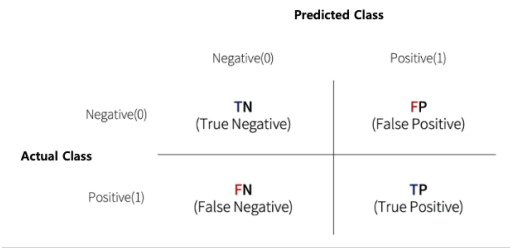

그림 4. 오차 행렬
Fig. 4 Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad \cdots (1)$$

True Positive (TP), True Negative (TN),
Fasle Positive (FP), False Negative (FN)

본 논문에서 모델 성능 평가는 그림 4의 오차 행렬로 식 (1)의 Accuracy 계산식으로 진행되었다. 약 100개의 테스트 데이터를 실제 클래스와 모델이 분류한 클래스로 비교하여 점수를 산정하였다. 식 (1)의 Accuracy는 평균 92%의 높은 정확도를 보였다.

5.2 초기화면

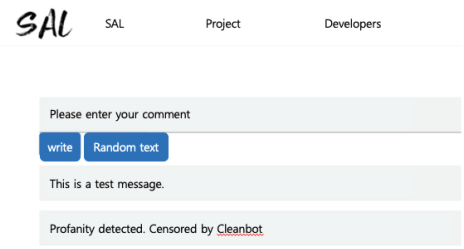

그림 5. 메인 페이지
Fig 5. main page

그림 5는 모델이 저장된 API서버와 통신할 수 있는 메인 페이지이다. 해당 페이지에 텍스트를 입력하면 API 서버와 통신 후 Clean으로 분류될 경우 작성한 내용을 이용자가 볼 수 있도록 댓글 박스에 데이터를 추가하고 리랜더링 되어 화면에 출력한다. 그 이외의 분류로 분류되었을 경우에는 어떤 분류에 속하는지 보여 주고 작성된 내용을 읽을 수 없도록 검열하여 화면에 랜더링 된다.

4.3 중간 처리 과정

그림 5. 메인 페이지에서 사용자가 입력한 텍스트는 POST BODY에 담겨 API 서버로 송신된다. API 서버는 Tokenizer 전에 텍스트의 타당성을 검사한다. “아이콘” 단독으로 사용하거나 문장에 섞을 경우 일부 집단을 비하하는 의미로 사용될 수 있다. “아이콘”이 제외된 텍스트는 Tokenizer을 거쳐 학습된 모델에서 연산된다. 연산된 값과 송신한 텍스트를 클라이언트에 반환한다.

Ⅵ. 결 론

본 논문에선 KoBert모델을 fine-tuning하여 비속어를 분류하는 모델을 기반으로 API 서버를 만들어 클라이언트 측에서 편리하게 사용할 수 있는 시스템을 구축했다. 한국어에 특성상 변형이 쉬워 다수의 웹 사이트 이용자가 무분별하게 비속어에 노출되는 것을 차단할 수 있다. 인터넷의 특성상 특정한 대상을 지칭하고 모욕성이 인정되면 사이버 모욕 범죄로 취급된다.

본 논문에선 사용한 KoBert는 LLM으로 API 서버에서 실행하면 메모리와 CPU에 많은 자원을 사용한다. API 서버에 사용자의 요청을 분산하여 처리해야할 필요가 있다. 유저의 요청을 분산 처리하는 기술은 로드 밸런싱이다. 서버에 할당하는 이용자의 수를 조절하면 자원 사용에 대한 이슈를 최소화 구현된 모델과 API 서버를 사용할 수 있다. 사용자가 많이 이용한다면 서버의 수를 증설하거나 클라우드 기반 시스템을 구현하여 비용을 줄일 수 있다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 지원을 받아 수행되었음 (2019-0-01817)

References

[1] S. Lee, I. Kang, Y. Jung, H. Kim, “Analysis of Cyber Crime and Its Characteristics,” *The Korea Society of Management Information Society*, vol 21, no 3, 2019, pp. 1-26. https://www.earticle.net/Article/A360323

[2] H. Park, “The Constitutional Study on the Hate Speech,” *Public Law J.*, vol. 16, no. 3, 2015, pp. 137-169. https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06513709

[3] J. Kim, “A Study on Fine-Tuning and Transfer Learning to Construct Binary Sentiment Classification Model in Korean Text,” *Proceedings of The Korea Society of Computer and Information Conference*, vol. 28, no. 5, 2023, pp. 15-30. https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11554934

[4] S. Lee, “A Swearword Filter System for Online Game Chatting,” J. of the Korea Institute of Information and Communication Engineering, vol. 15, no. 7, 2011, pp. 1531-1536.

https://doi.org/10.6109/jkiice.2011.15.7.1531

[5] S. Ko, Y. Shin, “Token-Based Classification and Dataset Construction for Detecting Modified Profanity,” *The Trans. of the Information Processing Society,* vol. 13, no. 4, 2024, pp. 181-188.
https://doi.org/10.3745/TKIPS.2024.13.4.181

[6] Janson Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Inchter, Fei Xia, Ed H. Chi, Quoc Le, Denny Zhou, “LLM-Based Class Diagram Derivation from User Stories with Chain-of-Thought Promptings,” In *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, New Orleans, USA, 2022

[7] G. Bea, C. Kim, S. Hwang, Y. Lee, J. Kang, “Development of Digital Exhibition Contents Using Generative AI and Prompt Engineering,” *J. of Korea Multimedia Society,* vol.27, no.8, 2024, pp. 959-968.
https://doi.org/10.9717/kmms.2024.27.8.959

[8] M. Lee, Y. Park, J. Na, C. Sohn "implement review sentiment analysis utilizing KoBERT, KoGPT-2, KoBART and handling hyperparameter optimization,” *J. of Digital Content Society,* vol. 24, no. 11, 2023, pp, 2831-2840.
https://doi.org/10.9728/dcs.2023.24.11.2831

[9] H. Cho, Y. Yi, H. IM, J. Cha, C. Lee, “Automatic Score Range Classification of Korean Essays Using Deep Learning-based Korean Language Models -The Case of KoBERT & KoGPT2-,” *J. of the Int. Society of Korean Language and Culture,* vol. 18, no. 1, 2021, pp. 217-241
. http://dx.doi.org/10.15652/ink.2021.18.217

[10] G. Kim, S. Park, M. Kwon, Erniyozov Shokhrukh, "Classifying Images of the ASL Alphabet using Dual homogeneous CNNs structure," *J. of The Korean Institute of Electronic Communication Sciences*, vol. 18, no. 3, 2023, pp. 449-458.

[11] J. Lee, "Comparison of RNN and Transformer-based Models for Sentiment Classification of Korean Reviews", *J. of The Korean Institute of Electronic Communication Sciences*, vol. 18, no. 4, 2023, pp. 693-700.

저자 소개

김영민(Young-Min Kim)

2024년 동서대학교 소프트웨어학과 졸업(공학사)

※ 관심분야 : 소프트웨어, 인공지능

박승민(Seung-Min Park)

2010년 중앙대학교 전자전기공학부 졸업(공학사)
2019년 중앙대학교 대학원 전자전기공학과 석박사통합과정 졸업(공학박사)
2019년~현재 동서대학교 소프트웨어학과 조교수
2022년~현재 동서대학교 AI+X융합연구센터장
2021년~현재 산업인공지능 표준화포럼 운영위원
※ 관심분야 : 인공생명, 패턴인식, 뇌-컴퓨터 인터페이스, 기계학습.