박 새 롬 교수 지도

석사학위 청구논문

순서형 회귀 모형을 활용한 인공지능 악성댓글탐지 모형의 성능 향상 연구

2023

성신여자대학교 대학원

미래융합기술공학과

이 세 영

순서형 회귀 모형을 활용한 인공지능 악성댓글탐지 모형의 성능 향상 연구

박 새 롬 교수 지도

이 논문을 석사학위논문으로 제출함

2022년 11월

성신여자대학교 대학원
미래융합기술공학과
이 세 영

인 준 서

이세영의 석사학위 논문으로 인준함

2022년 11월

심사위원장 김 경 진 (서명 또는 인)

심 사 위 원 박 새 롬 (서명 또는 인)

심 사 위 원 이 주 희 (서명 또는 인)

성신여자대학교 대학원

논문 개요

컴퓨터 통신 기술의 발달과 COVID-19 바이러스의 여파로 온라인에서의 활동이 활발해졌다. 특히 YouTube, Tiktok 과 같은 온라인 콘텐츠를 즐길 수 있는 SNS 활동이 급속도로 증가함에 따라 SNS 플랫폼에서 콘텐츠를 즐기고 자신의 의견을 온라인 댓글을 통해 표출하는 경우가 많아졌다. 온라인 특성상 익명이 보장되기 때문에 표현의 자유를 악용하여 혐오 발언 또는 편견 발언이 담긴 악성댓글이 작성되기 쉽다. 온라인 악성댓글은 오프라인에 실존하는 대상에게 정신적인 피해를 준다. 악성댓글로 인해 대상자가 극단적인 선택을 하는 경우가 발생할 수 있으므로 악성댓글에 대한 사전 방지책과 규제 방안이 절실하게 필요하다.

한국어 악성 댓글 분류 모델을 학습하기 위해 한국어 댓글 데이터를 수집한 데이터셋으로 KOCO(KOrean COmments) 데이터셋이 있다. KOCO 데이터셋 중 KOCO-hate 데이터셋은 악성 댓글을 혐오감의 정도에 따라 정상, 공격적인 발언, 심한 혐오 발언으로 레이블링을 수행하였다. 따라서 악성댓글 분류는 혐오 발언의 정도에 따른 다중 분류 문제이기 때문에 각 클래스의 순서 정보를 활용하기 위해 순서가 있는 클래스를 분류하는데 효과적인 순서형 회귀 모형을 활용한 악성 댓글 분류 모델을 제안한다. 먼저, 혐오 발언 분류를 위해서 사전학습된 한국어 자연어처리 모델에 순서형 회귀 모형인 CORAL(COnsistent RAnk Logits) 프레임워크와 CORN(Conditional Ordinal Regression for Neural network) 프레임워크를 악성 댓글 분류 모델에 적용하였다. 기본모형, CORAL 모형, CORN 모형의 분류 성능을 비교했을 때 순서형 회귀 모형을 활용한 CORAL과 CORN 모형에서 성능이 향상된 것을 확인하였다.

목 차

표 차 례

그림 차례

I. 서론

1. 연구 배경 및 목적

Social Network Service(SNS)는 서비스 사용자 간의 사회적 관계를 형성하고 의사소통 및 상호교류가 가능한 온라인 플랫폼을 말한다. 온라인은 오프라인보다 시공간적 제약이 적고, 자신의 신분을 노출하지 않으면서 익명으로 서비스를 이용할 수 있어 오프라인보다 자유롭게 활동할 수 있다. 인터넷과 통신기술의 발달로 고정된 장소에서 인터넷을 사용하던 PC 중심의 서비스에서 벗어나 이동 시에도 사용할 수 있는 모바일 기기 중심으로 서비스가 변화하면서 SNS를 통한 온라인 콘텐츠 소비가 매우 활발해졌다. 온라인 댓글은 사회 이슈 또는 온라인 콘텐츠에 대한 자신의 의견을 표출하는 창구로 사용된다.

온라인에서 보장되는 익명성에 숨어 특정인에 대해 심각한 욕설을 하거나 비방, 조롱하는 댓글을 남기기도 한다. 실존하는 특정인을 비방하거나 조롱하는 댓글일 경우, 이는 현실의 실존 인물에게 정신적인 피해를 주기도 한다[1]. 이러한 피해는 COVID-19 바이러스로 인한 팬데믹을 겪으면서 더욱 심각해졌다. 코로나 이후 특정 인종에 대한 혐오 댓글이 증가하였으며[2], COVID-19 바이러스 확산 방지를 위한 사회적 거리두기로 인해 외부 활동이 감소하여 인터넷 사용이 활발해지고 각종 온라인 콘텐츠의 생성과 소비가 늘어나면서 자연스럽게 온라인 댓글이 증가하였고, 이에 악성댓글 또한 증가하였다[3]. 악성댓글로 인한 피해를 사전에 방지하기 위해 인공지능을 활용한 악성댓글 탐지 연구가 활발하게 진행되고 있다.

한국어 악성댓글 분류 연구를 위해 구축된 데이터셋에는 KOCO(KOrean COmments)[1], Unsmile 데이터셋 등이 있다. KOCO 데이터셋은 한국어 연

예 뉴스 기사의 댓글을 수집하여 약 1만여 개의 댓글에 대해 혐오 발언의 정도에 따라 hate, offensive, none으로 레이블링을 수행하였고, 댓글 문장에 포함된 차별 발언의 유형에 따라 gender, others, none으로 레이블링이 되어 있다.

본 논문에서는 KOCO-hate 데이터셋의 분류 성능을 향상시키기 위한 방법을 제안한다. KOCO-hate 데이터셋에서는 혐오 발언의 정도가 명확하게 구분되는 레이블이 아니라, 혐오 발언이 심한 정도에 따라 순서가 있는 클래스임에 착안하여 순서형 척도가 있는 데이터셋을 분류할 때 사용하는 순서형 회귀 모형을 활용한 악성댓글 분류 모델을 제안한다. 한국어 대용량 데이터로 사전학습된 자연어처리 모델에 순서형 회귀 모형인 CORAL 프레임워크와 조건부확률 기반의 순서형 회귀 모형인 CORN 프레임워크를 활용하여 상대방에 대한 비하나 조롱 등 불쾌감을 줄 수 있는 공격적인 어조 댓글과 직접적인 욕설이나 모욕적인 혐오 댓글로 나누어 혐오 발언의 정도에 따른 차이를 학습하도록 하였다. 사전학습된 한국어 자연어처리 모델로는 KoBERT, KcBERT, KoELECTRA, KcELECTRA, KoGPT2를 사용하였으며, 순서형 척도를 고려하지 않고 정답 클래스를 예측하는 기본모형, 순서형 척도를 고려한 CORAL 모형과 CORN 모형의 분류 성능을 비교하였다. KOCO-hate 데이터셋의 validation 데이터셋을 사용하여 학습된 모델의 성능을 평가하였다. F1-score와 Accuracy 등 여러 성능 지표를 통해 성능을 확인하였다.

2. 논문 구성

본 논문의 구성은 다음과 같다. 2장에서는 한국어 악성댓글 분류를 위해 선행된 연구를 살펴보고 선행연구에서의 한계점과 본 연구의 차별점에 대해 논하고, 악성댓글 탐지 연구를 위해 제안된 한국어 악성댓글 데이터셋인

KOCO 데이터셋에 대해 설명한다. 그리고 3장에서는 사전학습된 한국어 자연어처리 모델을 소개한 뒤, 4장에서는 본 연구에서 제안하는 순서형 회귀 모형을 활용한 한국어 악성댓글 분류 모델의 구조를 설명한 후 5장에서 실험을 통해 분류 모형별 성능을 확인한다. 마지막으로 6장에서는 향후 연구 방향에 대해 논의하고 결론을 내린다.

II. 한국어 악성댓글 분류

1. 악성댓글 분류 연구

SNS 활동이 활발해지면서 온라인 콘텐츠와 댓글의 생성량이 급격하게 증가하였다. 이전에는 사이트 관리자가 악성댓글을 직접 검수하였지만, 기하급수적으로 악성댓글이 증가하여 인력으로 악성댓글 문제를 해결하는 것은 불가능해졌다. 따라서 인공지능을 활용하여 악성댓글을 탐지하고자 하는 연구가 활발히 진행되고 있다.

SVM(Support Vector Machine)과 감성 분석을 활용하여 악성댓글을 탐지한 연구에서는 단어의 악의성 수치를 0~1로 표현하고, 악의성 수치를 SVM 모델에 학습시켜 악성댓글을 탐지하고자 하였다[4]. 이전의 악성댓글 탐지 연구보다 좋은 성능인 재현율 87.8%를 보였지만, 단어사전 및 단어의 악의성 수치를 계속해서 업데이트하여야 한다는 한계점이 있었다. 본 연구에서는 주기적인 업데이트를 요구하지 않도록 한국어 문장의 문맥을 학습하여 분류하는 딥러닝 모델을 사용하여 악성댓글을 탐지하고자 한다.

KOCO 데이터셋과 심심이 나쁜 말 데이터셋을 활용하여 KoELECTRA 모델을 학습시킨 후 악성댓글 분류 성능을 확인한 연구에서는 각 데이터셋의 학습 후 성능이 F1-score 0.63, 0.66으로 낮은 성능을 보이는 이유에 대한 정성분석을 실시하였다[5]. 본 연구에서는 데이터셋의 한계점을 파악하고, KOCO 데이터셋이 가진 순서형 척도의 특성을 활용하여 순서형 회귀모형을 통해 분류 성능을 향상시키고자 한다.

한국어 악성댓글을 효과적으로 분류하기 위한 하이웨이 네트워크 기반 CNN(Convolutional Neural Network) 모형연구에서는 새로운 분류 모델을 제시하여 KOCO 데이터셋의 분류 성능을 향상시켰다[6]. OOV(Out Of

Vocabulary) 사전학습 임베딩 방식으로 신조어가 많은 댓글 데이터에 적합한 임베딩을 수행하였고, 하이웨이 네트워크 기반의 CNN 모형을 구축하여 KOCO-hate 데이터셋과 KOCO-bias 데이터셋을 함께 학습시켜 멀티레이블을 예측하도록 학습하였다. KOCO-hate validation 데이터셋의 F1-score는 0.62로 KOCO 연구팀의 베이스라인 모델인 KoBERT의 F1-score보다 향상된 성능을 보였다.

비정형데이터인 댓글 데이터는 사용되는 단어, 띄어쓰기 등의 문법에서 불규칙함을 보였다. 따라서 본 연구에서는 레이블 데이터에 주목하였다. 수학적으로 나타내기 비교적 간단하고, 자연어보다 규칙성있는 레이블 데이터를 활용하여 분류성능을 높이고자 하였다. 본 연구에서는 새로운 구조의 모델을 구축하지 않고, 사전학습된 자연어처리 모델의 출력 레이어를 변형하고 손실 함수를 변경하는 비교적 간단한 방법을 사용하여 선행된 연구보다 더 높은 성능을 보였다.

2. 악성댓글 데이터셋

KOCO 데이터셋은 한국어 온라인 연예 뉴스의 댓글 데이터를 수집하여 혐오 발언의 정도와 특정 집단에 대한 편견 또는 차별에 따라 레이블링한 데이터셋이다. 연예 뉴스 기사에 대한 댓글 데이터셋이기 때문에 데이터셋 내 혐오 발언 혹은 차별 발언의 대상이 연예인인 경우가 많았다.

[표 1] KOCO 데이터셋 예시

Comments	Hate	Bias
20년 연애 너무 멋져요 !! 이제 결혼하셔서 행복한 가정 이루시길	none	none
30대 아줌마들;; 고만하세요;	offensive	gender
52면 날씬한건데..	none	others
ㅋㅋㅋ 저런것도 정신병임. 정신과상담받으세요	hate	none
YG 제국 부활하자 빅뱅 컴백해서 방탄 밟자!	offensive	none
33살이 청년임? 아저씨아님?	hate	others

약 1만 개의 레이블링된 데이터와 약 200만 개의 레이블링이 되지 않은 댓글 데이터가 있다. 약 8천 개의 문장으로 구성된 Train 데이터셋, 약 5백 개의 문장으로 구성된 Validation 데이터셋, 그리고 약 1천 개의 문장으로 구성된 Test 데이터셋으로 구성되어 있다. Test 데이터셋의 레이블은 공개되지 않고, KOCO 연구팀에서 진행하고 있는 Kaggle competition에 예측 결과를 제출하여 예측 성능을 확인할 수 있다. KOCO 데이터셋의 예시 데이터는 표 1에서 확인할 수 있다.

KOCO-hate 데이터셋은 KOCO 데이터셋 중 혐오 발언의 정도에 따라 강한 증오심이나 모욕적인 내용의 댓글에는 hate 레이블, 대상이나 독자에게 불쾌감을 주거나 공격적인 내용의 댓글에는 offensive 레이블, 그리고 증오나 모욕적인 발언이 포함되지 않은 댓글에는 none 레이블을 부여한 데이터셋이다. KOCO-hate 데이터셋은 혐오의 정도에 따라 레이블을 부여했기 때문에 레이블에 혐오의 정도가 hate > offensive > none 순으로 크다고 할 수 있다.

KOCO 데이터셋의 레이블링은 KOCO 연구팀에서 제시한 가이드라인에

따라 32명의 작업자가 레이블링을 수행하였다. 레이블링의 신뢰성을 확보하기 위해 작업자들의 합의 지수를 나타내는 Krippendorff's alpha[7]를 확인한 결과, KOCO-hate 데이터셋에서는 0.496으로 보통 수준의 합의를 달성하였다. 이러한 데이터셋을 일반적인 클래스 분류 기법을 사용하여 분류하면 혐오의 정도에 대한 순서 정보를 활용할 수 없다. 따라서 이러한 순서 정보를 고려하여 분류작업을 수행한다면 분류성능이 향상될 것을 기대할 수 있다.

[표 2] KOCO-hate 데이터셋 구성

	offensive	hate	none	Total
train	2499	1911	3486	7896
validation	189	122	160	471
test	-	-	-	974

KOCO-bias 데이터셋은 KOCO 데이터셋 중 특정 집단에 대한 편견이나 차별에 따라 성과 관련된 편견 혹은 차별 발언이 포함된 댓글에는 gender 레이블, 특정 집단에 대한 편견 혹은 차별 발언이 포함된 댓글에는 others 레이블, 그리고 편견이나 차별적 발언이 포함되지 않은 댓글에는 none 레이블을 부여한 데이터셋이다. KOCO-bias 데이터셋 또한 데이터 레이블링의 신뢰성을 확보하기 위해 Krippendorff alpha를 계산한 결과, 0.492로 보통 수준의 합의를 달성하였다.

KOCO-bias 데이터는 차별적 발언이 들어간 gender, others 데이터의 합보다 none 데이터가 약 2배 더 많은 매우 불균형한 데이터셋이다.

KOCO-bias 데이터셋에서 성차별적 발언을 제외한 지역, 연령, 종교 등 다양한 편견이 모두 others 레이블로 레이블링이 수행되었고, 성차별적 발언과 다른 편견 발언이 함께 포함되었을 때 others와 gender 레이블로 multi-label 레이블링이 수행되지 않고, gender 레이블로만 레이블링이 수행되었다. KOCO-bias 데이터셋은 순서형 클래스를 가진 데이터셋이 아니므로 본 연구에서는 사용하지 않았다.

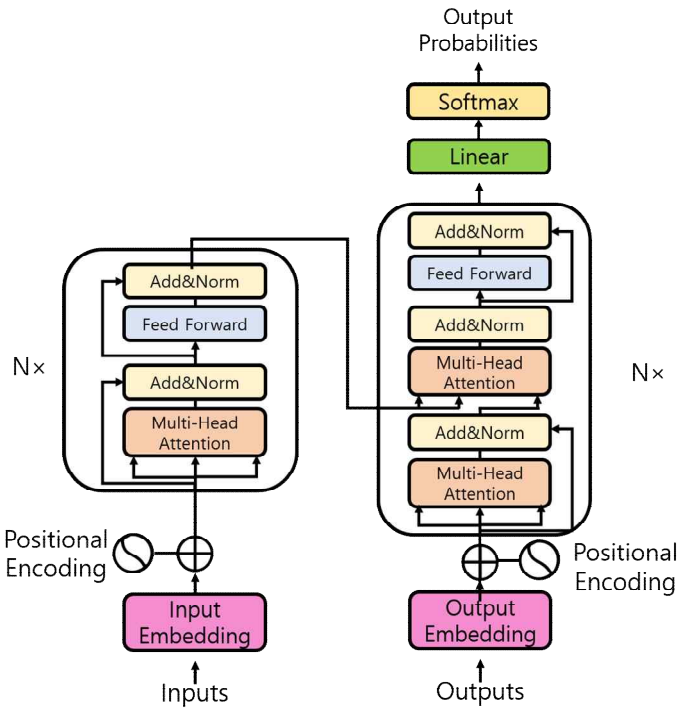
[표 3] KOCO-bias 데이터셋 구성

	gender	others	none	Total
train	1232	1516	5148	7896
validation	67	62	342	471
test	-	-	-	974

III. 한국어 자연어처리 모델

1. Transformer

Transformer는 Google에서 발표한 자연어처리 모델로 sequence to sequence 모델 기반의 encoder-decoder 구조를 가진 자연어처리 모델이다[8]. Transformer 모델은 Attention 기법을 활용하여 입력 데이터의 길이에 상관없이 정보를 학습할 수 있으며, 학습 시 디코더 모델 이후에 출력 레이어를 원하는 작업에 맞게 변경하여 학습할 수 있다. Linear layer를 추가하여 분류작업을 수행하거나, 디코더 모델을 연결하여 문장 생성 작업을 수행하는 등 다양한 작업에 활용할 수 있다. transformer 모델의 구조는 그림 1과 같다[8].

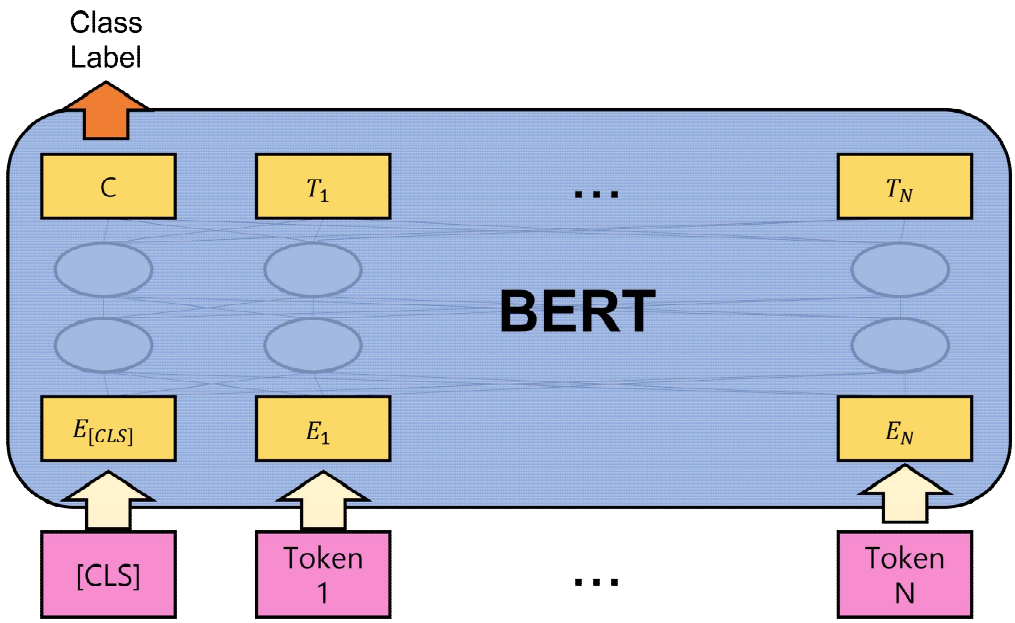


[그림 1] Transformer 모델 구조도

2. BERT 모델

BERT(Bidirectional Encoder Representation from Transformer) 모델은 Transformer 구조 기반의 자연어처리 모델로 Transformer 구조의 encoder를 여러 개 쌓아 올려 만든 구조이다[9]. BERT는 BookCorpus, English Wikipedia 등 영어로 된 대용량 말뭉치 데이터셋을 통해 사전학습을 수행하였다. 사전학습으로 두 가지 Unsupervised Learning 작업을 수행하여 모델이 자연어의 문맥을 학습하도록 하였다. 첫 번째 사전학습작업은 Masked Language Modeling 작업이다. 토크나이저를 사용하여 입력 문장을 토큰 시퀀스 형태로 처리한 뒤, 입력 토큰 시퀀스의 일부 토큰을 [MASK] 토큰으로 대체하고, 모델이 [MASK] 토큰을 원본 토큰으로 복원하는 방법으로 학습한다. 두 번째 사전학습작업은 Next Sentence Prediction 작업이다. 두 문장을 하나의 입력 문장으로 구성하고, [SEP] 토큰 전에는 첫 번째 문장을, 뒤에는 두 번째 문장을 배치한다. 그리고 token_type_ids를 두어 문장에 순서에 따라 토큰을 다르게 주어 문장의 순서 정보를 학습할 수 있도록 하였다. BERT 모델은 이 두 가지 사전학습 작업을 통해 문맥을 학습한다.

본 논문에서는 KOCO-hate 데이터셋의 댓글 데이터를 none, offensive, hate 3개의 클래스로 분류하기 위해 그림 2와 같이 BERT 모델에 입력 문장을 주면, 해당 문장의 클래스를 예측하도록 모델을 구성하기 위해 BERT 모델의 마지막 레이어를 classifier로 구성하였다. 기존의 BERT 모델은 이어지는 두 개의 문장을 입력으로 사용했지만, 본 논문에서는 온라인 댓글 특성상 한 단어로 이루어지는 댓글이 있을 정도로 길이가 매우 짧고, 대부분 한 문장으로 이루어지는 경우가 많았다. 데이터 전처리 과정에서 두 문장 이상으로 구성된 문장이라도 한 문장으로 취급하였으며, 문장의 순서를 학습할 수 있는 token_type_ids는 사용하지 않았다.



[그림 2] BERT classification 모델 구조도

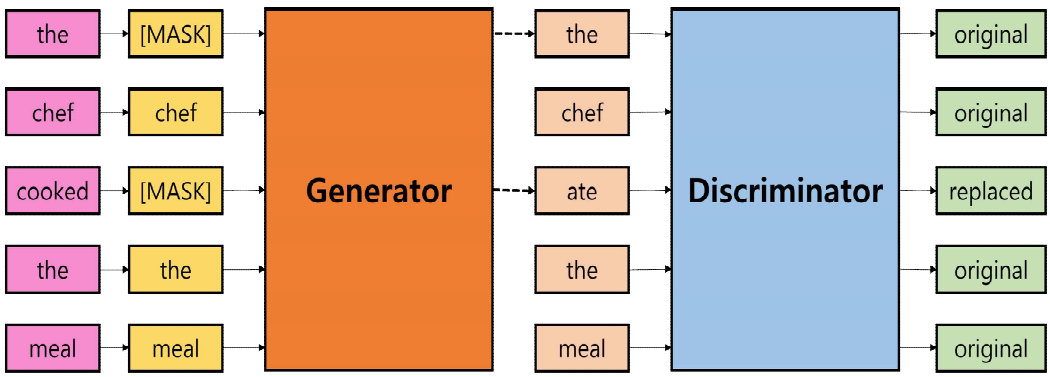
BERT 모델에 대용량 한국어 데이터셋을 학습시킨 모델로 KoBERT 모델이 있다[10]. SKT에서 BERT 모델에 한국어 위키피디아, 뉴스 기사 등의 대용량 한국어 말뭉치 데이터셋을 사용하여 사전학습을 수행한 한국어 자연어처리 모델이다. 한국어 자연어처리 모델의 초기 모델인 KoBERT의 단어사전 크기는 8,002개로 다른 자연어처리 모델들보다 비교적 작은 크기의 단어사전을 가지고 있다.

KcBERT 모델은 BERT 모델에 한국어 댓글 데이터셋을 사용하여 사전학습을 수행한 모델이다[11]. 온라인 댓글 특성상 문장 길이가 짧고, 정형화되지 않은 데이터가 많아 모든 한국어 데이터에 대해 성능이 매우 좋은 것은 아니지만, 다른 한국어 자연어처리 모델과 비교했을 때, 댓글 데이터와 같이 신조어, 비속어, 줄임말 등이 많은 한국어 비정형 데이터에 대한 작업을 수행했을 때, 성능이 비교적 좋다. KcBERT의 단어사전 크기는

54,343개로 KoBERT에 비해 매우 큰 크기의 단어사전을 가지고 있다.

3. ELECTRA 모델

ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately) 모델은 BERT 모델을 기초로 하여 설계된 모델이다[12]. BERT 사전학습 작업 중 하나인 Masked Language Modeling 작업을 변형한 Replaced Token Detection 작업을 통해 사전학습을 수행하여 BERT 보다 자연어처리 성능을 향상시켰다. ELECTRA 모델은 그림 3과 같이 Generator와 Discriminator로 구성되어 있다[12]. ELECTRA 모델의 사전학습 방법인 Replaced Token Detection 작업은 다음과 같이 수행한다. 입력 문장의 일부를 [MASK] 토큰으로 치환한 후 Generator에 입력한다. Generator는 입력 문장의 [MASK] 토큰을 다시 [Replaced] 토큰으로 변환하여 Discriminator에 전송한다. Discriminator는 입력받은 문장의 토큰 중 치환된 토큰을 찾고, 해당 토큰을 원본 토큰으로 복원하는 작업을 수행한다. 적대적 학습과 유사한 사전학습과정을 통해 문맥 이해를 높여 자연어처리 성능을 높인 모델이다.



[그림 3] ELECTRA 모델 구조도

KoELECTRA 모델은 ELECTRA 모델에 대용량 한국어 데이터셋을 통해 사전학습 시킨 모델이다[13]. 한국어 뉴스 기사 데이터셋, 한국어 위키피디아, 그리고 국립국어원에서 구축한 모두의 말뭉치 데이터셋을 학습시켜 정형화된 한국어 데이터에 대해 좋은 성능을 보이는 모델이다. 또한, 단어사전 크기는 35,000개이며, KoBERT보다 약 4배정도 큰 단어사전을 가지고 있다.

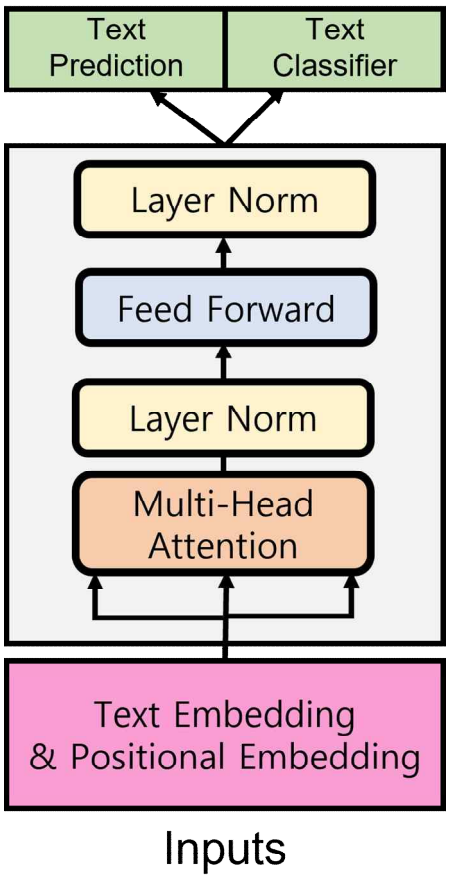
KcELECTRA 모델은 ELECTRA 모델에 한국어 댓글 데이터를 사용하여 사전학습 시킨 모델이다[14]. 한국어 댓글 데이터를 사용하였기 때문에 신조어, 오탈자 등이 많이 포함되어있는 비정형 한국어 데이터에 대해 KoELECTRA, KoBERT 등 다른 모델보다 좋은 성능을 보인다. 단어사전의 크기는 54,343개로 KcBERT와 같은 크기의 단어사전을 가지고 있다.

4. GPT Model

GPT(Generative Pre-trained Transformer) 모델은 Transformer 기반의 모델로, OpenAI가 기계번역을 위해 개발한 모델이다[15]. 양방향(Bidirectional) 모델인 BERT와 달리, GPT 모델은 단방향(Unidirectional) 모델이다. GPT 모델은 사전학습 작업으로 Autoregressive decoder를 사용하여 이전의 단어를 통해 다음 단어를 예측하는 작업을 수행하였다. 문맥보다는 다음 단어를 잘 예측하도록 학습된 언어모델이기 때문에 문장 생성 등의 작업에 최적화되어있다. GPT-2모델은 GPT모델의 후속 모델로 GPT보다 좋은 성능을 보인다[16]. GPT-2모델은 OOV(Out Of Vocabulary)문제에 더 유연하게 대처하기 위해 byte-pair-encoding을 사용하였다. 또한, GPT-2 모델은 GPT모델보다 더 많은 데이터셋을 학습하였고, 더 큰 크기의 단어사전을 가진다.

KoGPT-2 모델은 AWS와 SKT가 협력하여 GPT-2 모델에 한국어 데이

터셋을 사용하여 사전학습 시킨 한국어 자연어처리 모델이다[17]. 한국어 위키백과, 뉴스 기사, 모두의 말뭉치 v1.0, 청와대 국민청원 데이터셋 등의 한국어 대용량 데이터셋으로 학습되었다. BERT 기반의 한국어 자연어처리 모델보다 분류작업 성능은 낮지만, 문장 생성에서 좋은 성능을 보인다.



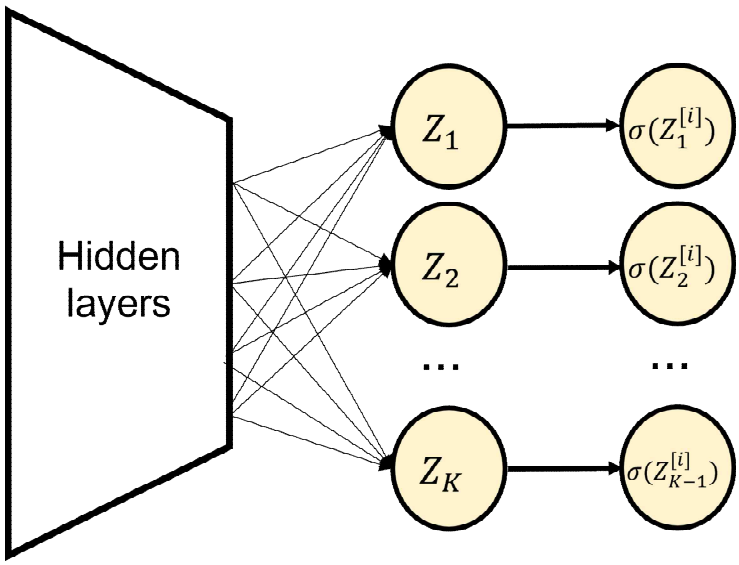
[그림 4] GPT 모델 구조도

IV. 악성댓글 분류 모델 설계

1. Ordinal Regression

Ordinal Regression이란 나이 예측, 손상 평가 등의 순서가 있는 척도를 가지는 다중 클래스 분류 문제에서 활용되는 분류 기법이다[18].

기본적인 순서형 회귀 모형은 그림 5와 같이 모델의 마지막 레이어에서 클래스 수보다 하나 적은 개수만큼의 이진 분류기를 두고, 각 클래스에 해당하는 확률을 sigmoid 함수로 계산한다. 분류기별로 계산된 확률값이 임계치를 넘는 분류기의 수를 세어 더한 뒤, 입력 데이터에 해당하는 클래스를 예측한다. 이후 이진 분류기들이 예측한 클래스 순서가 일관적이지 않은 경우를 대비하여 SVM 등의 기법으로 다시 순서를 Re-ranking 하여 최종적으로 예측한다. 하지만, 이 경우 Re-ranking 단계는 딥러닝 기법으로 학습이 되지 않아 순서에 따른 일관성 있는 예측을 보장할 수 없다. 이러한 순서 일관성을 보장하기 위해 CORAL이 제안되었다[19].



[그림 5] Ordinal Regression 모형 구조도

2. CORAL

CORAL(**CO**nsistent **RA**nk **L**ogits) 프레임워크는 순서형 회귀 모형에서 일관성 있는 분류를 위해 제안되었다[19]. 얼굴 사진을 통해 나이를 예측하는 성능을 향상시키기 위해 처음 고안되었으며, 이후 병의 진행 정도를 예측하거나[20], 감정인식에 활용하는 등[21] 순서형 척도를 가지는 다양한 분류 문제에 사용되어 좋은 성능을 보였다.

CORAL은 분류하고자 하는 클래스의 수를 K라고 하면, 예측 변수 Y의 클래스를 순위에 맞게 정렬하고, 예측 변수 y_i에 해당하는 순위 r_k로 표현한다. r_k로 표현된 y_i는 K-1 개의 벡터로 확장하여 $y_i^{(1)}, ..., y_i^{(K-1)}$로 나타낸다.

$$y_i \in Y = r_1, r_2, ..., r_k \ (r_k > r_{k-1} > ... > r_1) \tag{1}$$

$$\begin{aligned} y_i^{(k)} &\in \{0,1\}, \\ y_i^{(k)} &= 1 \cdot \{y_i > r_k\} \end{aligned} \tag{2}$$

이때, $1\{\cdot\}$ 함수는 내부 조건이 참이면 1을, 거짓이면 0을 반환하는 함수이다. 예를 들어, 클래스 수가 5이고, 클래스 5에 해당하는 데이터가 있을 때, 예측 변수 Y={1,1,1,1} 으로 표현할 수 있다.

CORAL은 마지막 레이어를 K-1 개의 이중 분류기로 구성하여 일관성 있는 순서형 척도로의 분류를 보장한다.

$$\sum_{i=1}^{N}\sum_{k=1}^{K-1}\lambda^{(k)}\begin{bmatrix}\log(\sigma(g(x_i, W) + b_k))y_i^{(k)} \\ + \log(1-\sigma(g(x_i, W) + b_k))(1-y_i^{(k)})\end{bmatrix} \tag{3}$$

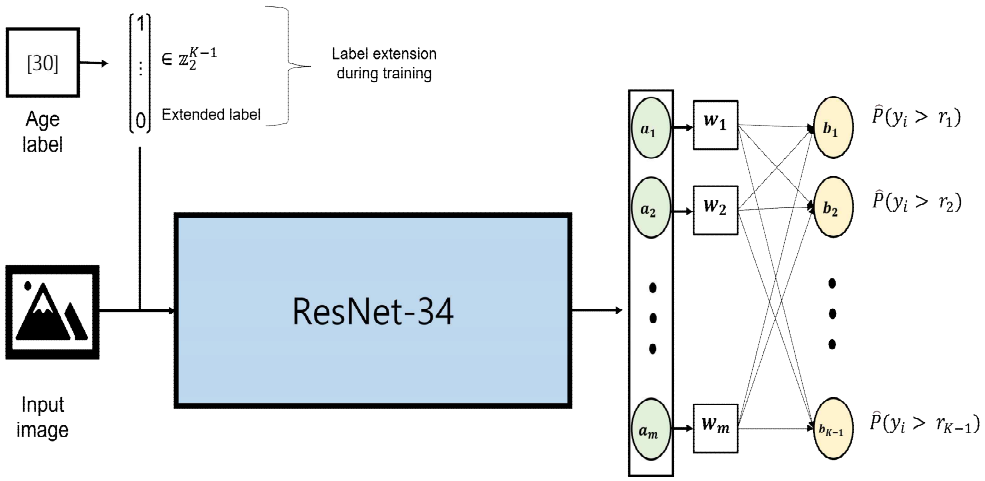
CORAL은 K-1개의 이진 분류기를 사용하기 때문에 binary cross-entropy loss와 유사한 손실 함수를 사용한다. 수식 3은 데이터 N개에 대하여 i번째 입력 데이터인 x_i 데이터의 k번째 이진 분류기에서의 loss를 구하는 수식이다. λ^k는 k번째 분류기에 대한 중요도 가중치를 나타내는 하이퍼파라미터이다. $g(\bullet)$ 함수는 최종 출력층의 직전 레이어에서 출력을 구하는 함수이며, $y_i^{(k)}$는 예측 변수 Y의 i번째 데이터의 k번째 벡터를 의미한다.

분류 모델을 $h(x)$라고 하면, 최종 예측은 $h(x^{[i]}) = r_q$로 구해지며, $q \in \{1, 2, ..., K\}$ 는 순위에 해당하는 숫자를 나타낸다. q를 구하는 식은 다음과 같다.

$$q = 1 + \sum_{k=1}^{K-1} f_k(x^{[i]}) \tag{4}$$

$f_k(x^{[i]}) \in [0,1]$ 은 k번째 이진 분류기의 출력계층에서 예측된 확률 값을 나타낸다. 그리고, $1\{\bullet\}$함수는 내부 조건이 참이면 1, 거짓이면 0을 나타내는 함수이다.

CORAL은 $\{f_k\}_{k=1}^{K-1}$에 대한 순위 일관성을 보장한다. 즉, $f_1(x^{[i]}) \geq f_2(x^{[i]}) \geq ... \geq f_{K-1}(x^{[i]})$ 의 순서를 보장한다.



[그림 6] CORAL 프레임워크 구조

3. CORN

CORN(**C**onditional **O**rdinal **R**egression for **N**eural Networks) 프레임워크는 조건부확률 기반의 순위 일관성을 보장하는 순서형 회귀 모형이다[22]. CORN은 CORAL과 유사하지만, 조건부확률을 활용하여 유연성과 표현력을 높인 모형이다. 분류하고자 하는 클래스의 수가 K개라고 한다면, CORN에서도 CORAL과 같은 방식으로 예측 변수 Y의 클래스를 순위에 맞게 정렬한 후 예측 변수 y_i를 순위 r_k로 표현하고, y_i는 수식 2에 따라 $y_i^{(k)}$로 확장하여 나타낸다. K-1개의 이진 분류기의 출력값을 최종 예측 레이블인 q는 수식 5와 같이 구한다.

$$q = 1 + \sum_{k=1}^{K-1} 1\left\{f_k\left(x^{[i]}\right) > 0.5\right\} \tag{5}$$

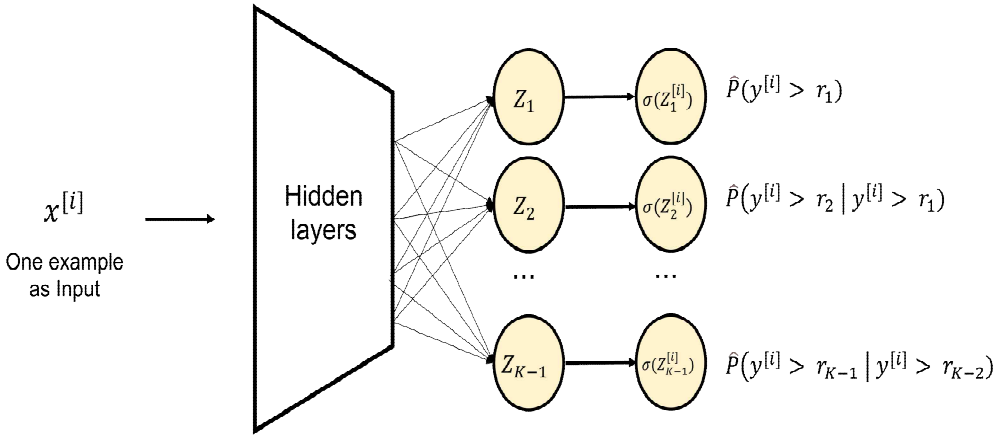
CORN 모형에서 확장된 레이블 $y_i^{(k)}$를 구하는 식 $f_k(x^{[i]})$는 CORAL 모형과 달리 수식 6과 같이 조건부확률의 곱으로 계산한다.

$$f_k(x^{[i]}) = \hat{P}(y^{[i]} > r_1) = \prod_{j=1}^{k} f_j(x[i]) \tag{6}$$

$$\begin{gathered} S_1 : all\{(x^{[i]}, y^{[i]})\}, \text{for } i \in \{1,...,N\}, \\ S_2 : \{(x^{[i]}, y^{[i]}) | y^{[i]} > r_1\} \\ S_{K-1} : \{(x^{[i]}, y^{[i]}) | y^{[i]} > r_{k-2}\} \end{gathered} \tag{7}$$

$$L(x,y) = -\frac{1}{\sum_{j=1}^{K-1} |S_j|} \sum_{j=1}^{K-1} \sum_{i=1}^{|S_j|} \begin{bmatrix} \log(f_j(x^{[i]}) \cdot 1\{y^{[i]} > r_j\} \\ + \log(1 - f_j(x^{[i]})) \cdot 1\{y^{[i]} \le r_j\} \end{bmatrix} \tag{8}$$

손실 함수는 수식 8과 같이 CORAL과 유사한 식을 활용하지만, 조건부확률을 기반으로 하여 구한다. 수식 7에서 데이터의 레이블에 따라 조건부 집합으로 나누어 각 집합의 원소 수를 의미하는 $|S_j|$를 구한다. 조건부확률 기반의 손실 함수를 계산하여 손실 함수를 최소화하고, 입력 데이터의 실제 클래스와 예측 클래스의 차이가 작아지도록 학습한다. CORN은 손실함수에서 CORAL과 달리 하이퍼파라미터를 두지 않는다.



[그림 7] CORN 프레임워크 구조

V. 실험 결과

1. 실험 환경

본 연구에서는 사전학습된 한국어 자연어처리 모델에 CORAL과 CORN 프레임워크를 사용했을 때 혐오 발언 분류 성능을 측정하고 제안하는 모델과 비교 모델의 성능 비교 및 평가를 위해 Python 3.6.3 환경에서 모델을 직접 구축하여 학습한 후 성능을 평가하였다. 모델 구축 및 성능 평가는 GPU를 활용한 시스템 환경에서 수행되었다. 각 모델별 상세 실험 환경은 표 4에서 확인할 수 있다. 다만, KoBERT 모델의 경우, 사전학습된 모델과 transformers 라이브러리의 호환성 문제로 transformers 3.0.0 버전을 사용하였다.

[표 4] 실험 환경

	구분	명세
개발 환경	운영체제	Window 10 Pro 64bit
	CPU	Intel(R) Core(TM) i7-10700KF
	RAM	64GB
	GPU	NVIDIA GeForce RTX 3080
	메인보드	ROG STRIX Z490-G GAMING (Wi-Fi)
개발 언어 및 라이브러리	언어	Python 3.6.3
	transformers	Transformers 4.9.0
	torch	torch 1.7.1+cu110
	CUDA version	CUDA V11.5

본 연구에서 제안하는 모델과 비교 모델의 성능을 비교하기 위해 성능

지표로 Accuracy, F1-score, macro-F1, MAE, MSE 를 사용한다. KOCO-hate 데이터셋의 분류 성능 측정을 위해서 Accuracy, F1-score, Mean Absolute Error(MAE)와 Mean Squared Error(MSE)를 사용한다. KOCO-hate 데이터셋은 예측 변수로 여러 개의 클래스 중 데이터가 해당하는 하나의 클래스 값을 갖는 멀티클래스 데이터셋이다. 따라서 정답 클래스를 잘 맞추는지 확인할 수 있는 Accuracy와 불균형 데이터셋일 때 클래스별 데이터 수를 고려하여 성능을 측정할 수 있는 precision과 recall을 조합한 성능 지표인 F1-score를 활용하여 성능을 측정한다. F1-score에는 micro-F1과 macro-F2가 있다. micro-F1은 각 클래스별 정답 비율에 따른 평균값이고, macro-F1은 클래스별 샘플 수를 고려하여 구한 평균이다. 이진 분류 결과가 표 5과 같을 때, Accuracy와 F1-score, 그리고 MAE와 MSE를 구하는 식은 각각 수식 9와 수식 10이다. 표 5의 TP(True Positive), FP(False Positive), TN(True Negative), FN(False Negative)은 이진 분류의 결과에서 각각의 의미하는 바는 실제 값이 참일 때 예측 값도 참인 경우, 실제 값이 거짓인 경우 예측 값이 참인 경우, 실제 값이 거짓일 때 예측 값도 거짓인 경우를 의미한다. 데이터셋의 레이블이 K개라고 할 때, micro-F1과 macro-F1을 구하는 식은 각각 수식 11과 수식 12이다. 수식 11에서 TP_k, FP_k, FN_k 는 각각 k번째 클래스의 TP, FP, FN 을 의미하는 것이고, 수식 12에서 $Precision_k, Recall_k$는 k번째 클래스의 $Precision$과 $Recall$을 의미하는 것이다. 더불어 본 연구에서 제안하는 모델은 순서형 회귀 모형인 CORAL과 CORN을 사용하므로 회귀 모형의 성능 지표인 MAE와 MSE를 사용한다. MAE는 예측값과 실제값의 차이의 절댓값에 대한 평균을 취한 값이고, MSE는 예측값과 실제값의 차이의 제곱에 대한 평균을 취한 값이다. 이 지표가 작은 값을 가지는 것은 모델이 잘못 예측하더라도 hate 레이블을 가진 혐오 발언을 none 레이블인 정상 댓글로 예측하는 등

거리가 먼 클래스로 예측하지 않고, 거리가 가까운 클래스로 예측한다는 의미이다.

[표 5] 이진 분류 오차 행렬

		실제값	
		True	False
예측값	Positive	TP (True Positive)	FP (False Positive)
	Negative	TN (True Negative)	FN (False Negative)

$$
\begin{aligned}
Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\
\mathrm{Pr}ecision &= \frac{TP}{TP+FP} \\
Recall &= \frac{TP}{TP+FN} \\
F1score &= 2 \cdot \frac{1}{\frac{1}{Recall}+\frac{1}{\mathrm{Pr}ecision}} = 2 \cdot \frac{\mathrm{Pr}ecision \cdot Recall}{\mathrm{Pr}ecision + Recall}
\end{aligned}
\tag{9}
$$

$$
\begin{aligned}
MAE &= \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \\
MSE &= \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2
\end{aligned}
\tag{10}
$$

K개의 클래스가 있는 다중 분류에서는 각 클래스별로 이진 분류 문제를 풀듯이 TP, FP, FN, TP를 구하여 수식 9와 10을 따라 Accuracy 등의 평가 지표를 구한다.

KOCO-hate 데이터셋을 사용하여 한국어 자연어처리 모델들을 학습한

뒤 분류성능을 확인하였다. none, offensive, hate 모든 클래스에 대해 Precision, Recall 등의 성능 지표를 구하고, Accuracy, F1-score, MAE, MSE를 구하여 성능을 확인하였다. KOCO-hate 데이터셋은 offensive, hate 클래스의 데이터보다 none 클래스에 해당하는 데이터가 더 많은 불균형 데이터셋이므로 F1-score는 클래스별 샘플 수를 고려할 수 있는 macro F1-score를 사용하였다.

$$
\begin{aligned}
Micro-\mathrm{Precision} &= \sum_{k=1}^{K} \frac{TP_k}{TP_k + FP_k} \\
Micro-Recall &= \sum_{k=1}^{K} \frac{TP_k}{TP_k + FN_k} \\
Micro-F1 &= 2 \cdot \frac{1}{\frac{1}{Micro-Recall} + \frac{1}{Micro-\mathrm{Precision}}} \\
&= 2 \cdot \frac{Micro-\mathrm{Precision} \cdot Micro-Recall}{Micro-\mathrm{Precision} + Micro-Recall}
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
Macro-\mathrm{Precision} &= \frac{\sum_{k=1}^{K} \mathrm{Precision}_k}{K} \\
Macro-Recall &= \frac{\sum_{k=1}^{K} Recall_k}{K} \\
Macro-F1 &= 2 \cdot \frac{1}{\frac{1}{Macro-Recall} + \frac{1}{Macro-\mathrm{Precision}}} \\
&= 2 \cdot \frac{Macro-\mathrm{Precision} \cdot Macro-Recall}{Macro-\mathrm{Precision} + Macro-Recall}
\end{aligned}
\tag{12}
$$

2. 실험 모델 구성

1) 데이터 전처리

데이터 전처리는 모델별로 다르게 진행되었다. 각 모델의 미리 학습된 토크나이저를 활용하여 입력 문장을 토큰화하였고, 입력 문장의 최대 길이는 64로 설정하였다. 단어사전 외의 단어는 UNK 토큰으로 치환하였다. 모델의 입력으로는 토큰화된 문장인 input_ids, attention_mask, 실제 클래스 값인 label을 사용하여 학습을 수행하였다. 본 연구에서 사용한 데이터셋의 특성상 온라인 댓글은 두 문장 이상으로 이루어진 경우가 거의 없어, 다음 문장에 대한 정보를 알 수 있는 token_type_ids를 사용하지 않았다. 성능 평가를 진행할 때는 input_ids, attention_mask만을 사용하여 예측을 수행하였다.

[표 6] 모델별 데이터 전처리 설정

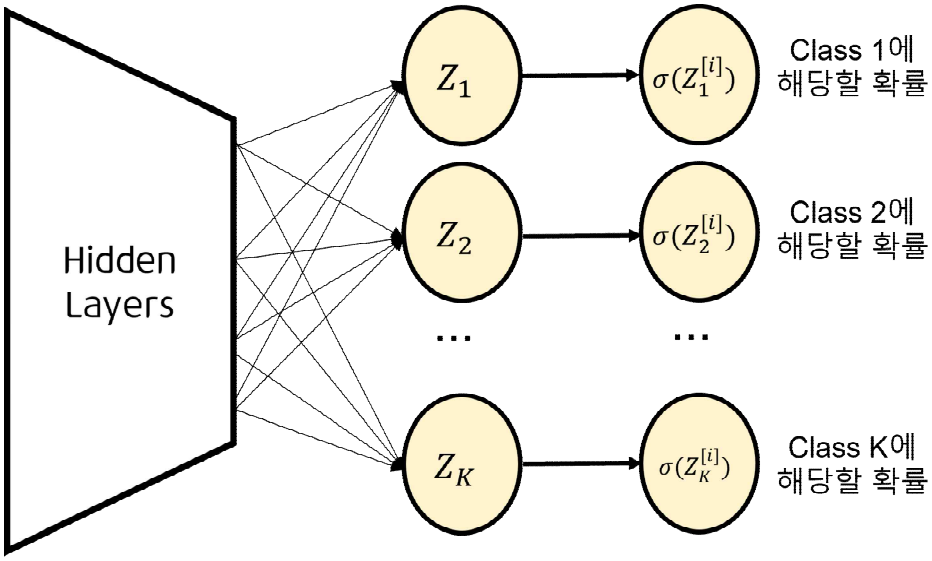
	토크나이저	단어사전 크기	문장 최대 길이
KoBERT	Bert 토크나이저	8,002	64
KcBERT	Bert 토크나이저	54,343	64
KoELECTRA	Electra 토크나이저	35,000	64
KcELECTRA	Electra 토크나이저	54,343	64
KoGPT-2	GPT-2 토크나이저	51,200	64

2) 실험 모델 구성

자연어처리 모델별 혐오 발언 분류 모형은 기본 분류 모형, CORAL 분류 모형, CORN 분류 모형으로 구성하고, 모형별 결과를 확인하였다.

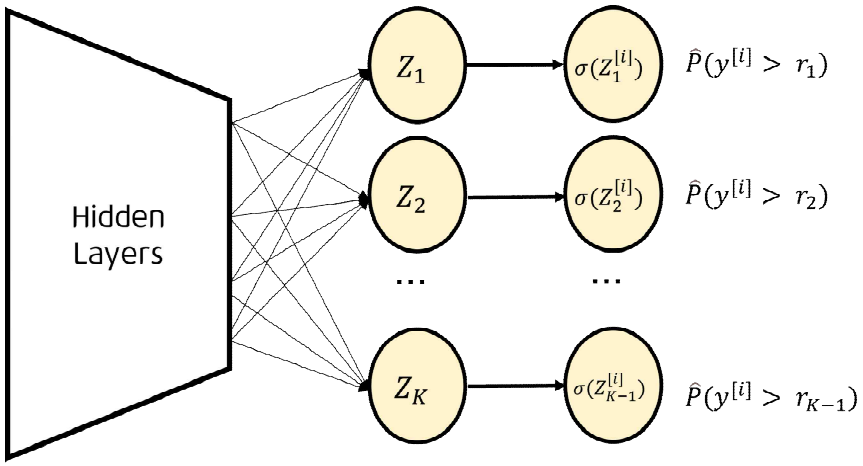
먼저, 기본 분류 모형은 사전학습된 자연어처리 모델의 출력 레이어를 클래스의 수만큼의 분류기로 구성하고, 각 클래스에 해당할 확률을 계산하

여 확률값이 가장 큰 클래스로 최종 예측을 수행하였다. 기본 분류 모형은 CrossEntropyLoss 함수를 손실함수로 사용하여 KOCO-hate train 데이터 셋을 사용하여 fine-tuning하였다. 기본 분류 모형의 구조는 그림 8과 같다.



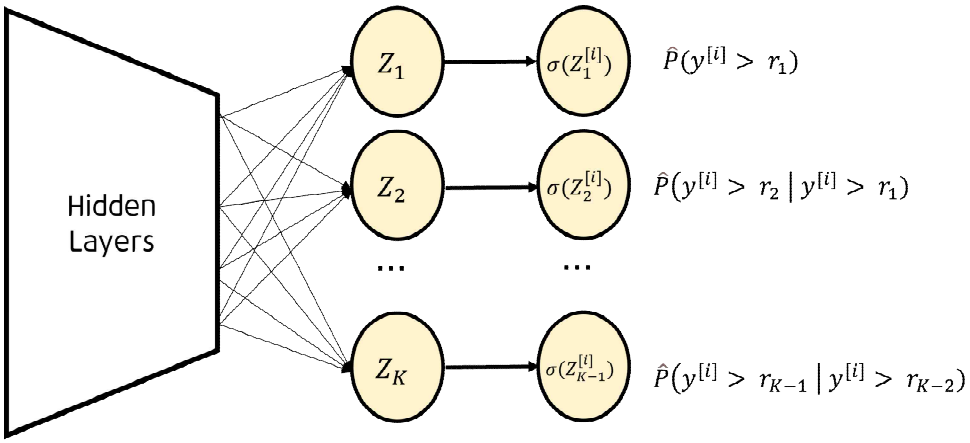
[그림 8] 기본 분류 모형 구조도

다음으로 CORAL 분류 모형은 사전학습된 자연어처리 모델의 출력 레이어를 클래스 수보다 한 개 적은 수의 분류기로 구성하였다. 다시 말해, 클래스 수가 K개라고 할 때, K-1개의 이진 분류기로 구성하여 각 이진 분류기마다 해당할 확률을 구하였다. 최종 예측을 할 때는 기본 분류 모형과 달리 가장 높은 확률값을 가지는 클래스로 예측을 하는 것이 아니라, 3장의 수식 4와 같이 각 분류기마다의 확률 중 0.5 이상의 확률을 가지는 분류기 수를 세어 예측 클래스를 구한다. KOCO-hate train 데이터를 사용하여 재학습을 수행할 때, 손실함수로는 CORAL 손실함수를 사용하여 학습하였다. CORAL 분류 모형의 구조는 그림 9과 같다.



[그림 9] CORAL 분류 모형 구조도

마지막으로 CORN 분류 모형은 CORAL과 마찬가지로 사전학습된 자연어처리 모델의 출력 레이어를 예측할 클래스의 수보다 한 개 적은 수의 분류기로 구성하였다. CORAL과 같이 클래스의 수가 K개라고 할 때, K-1개의 이진 분류기로 구성하여 각 이진 분류기마다의 확률값을 계산하였다. 3장의 수식 5와 같이 0.5 이상의 확률값을 가지는 분류기의 수를 구해 최종 예측을 수행하였다. KOCO-hate train 데이터를 모델에 재학습시킬 때, 손실함수로 조건부확률 기반의 함수인 CORN 함수를 사용하여 학습하였다. CORN 분류 모형의 구조는 그림 10과 같다.



[그림 10] CORN 분류 모형 구조도

3. 혐오 발언 분류 결과

KOCO-hate 데이터셋의 혐오 정도에 따른 순서 정보를 활용하여 분류 하기 위해 사전학습된 한국어 자연어처리 모형으로 KoBERT, KcBERT, KoELECTRA, KcELECTRA, KoGPT-2를 사용하여 성능을 비교하였다.

성능 비교를 위해 모델별 기본 분류 모형, CORAL 분류 모형, CORN 분류 모형을 구현하였다. 각 모델에는 KOCO-hate train 데이터셋을 학습시킨 후 validation 데이터를 사용하여 성능을 확인한 결과이다. 성능 지표로는 정확도(Accuracy), macro-F1, 평균절대오차(MAE), 평균제곱오차(MSE)를 사용하여 성능을 확인하였다.

1) 기본 분류 모형 성능

기본 분류 모형의 성능을 비교했을 때, 한국어 댓글 데이터로 학습시킨 KcELECTRA 모형이 Accuracy, macro-F1, MAE, MSE 모든 지표에서 가장 높은 분류성능을 보였다. 모델별 기본 분류 모형의 성능은 표 7에서 확인할 수 있다.

[표 7] 기본 분류 모형의 KOCO-hate 분류성능

	KoBERT	KcBERT	KoELECTRA	KcELECTRA	KoGPT2
Accuracy	0.60	0.61	0.64	**0.68**	0.57
F1-score	0.59	0.61	0.63	**0.68**	0.56
MAE	0.42	0.41	0.37	**0.33**	0.48
MSE	0.47	0.47	0.40	**0.35**	0.58

2) CORAL 분류 모형 성능

CORAL 분류 모형에서는 특정 클래스에 대한 중요도 가중치로 Importance Weights(iw) λ^k를 주어 클래스별 중요도를 조절하여 특정 클래스를 더 잘 분류하도록 모델을 학습시킬 수 있다. 1부터 K-1까지의 중요도 가중치를 $[\lambda^1,...,\lambda^{K-1}]$형태의 벡터로 설정하고, CORAL loss에서 이 하이퍼파라미터를 사용하여 계산한다. 본 연구에서는 기본값인 iw=[1, 1]과 offensive 레이블을 더 중요하게 분류하는 iw=[0.7, 0.3], 그리고 hate 레이블을 더 중요하게 분류하는 iw=[0.3, 0.7]로 설정하여 학습시켰다.

[표 8] CORAL 분류 모형의 KOCO-hate 분류성능

	KoBERT iw=[1,1]	KcBERT iw=[1,1]	KoELECTRA iw=[0.7,0.3]	KcELECTRA iw=[1,1]	KoGPT2 iw=[1,1]
Accuracy	0.60	0.62	0.65	**0.68**	0.57
F1-score	0.59	0.62	0.65	**0.68**	0.57
MAE	0.43	0.40	0.36	**0.32**	0.47
MSE	0.50	0.44	0.39	**0.33**	0.57

표 8의 각 모델별 성능은 3가지 iw를 사용하여 validation 데이터의 분류 성능을 확인했을 때, 가장 성능이 높았던 결과이다. 기본모형보다 성능이 향상되었으면 밑줄을 표시하였고, 가장 뛰어난 성능을 보였으면 글씨를 진하게 표시하였다.

3) CORN 분류 모형 성능

CORN 분류 모형은 CORAL 분류 모형과 달리 bias term에 영향을 미치는 importance weights를 사용하지 않고, 조건부확률을 기반의 손실 함수를 통해 학습한다. 표 9은 CORN 분류 모형의 KOCO-hate의 분류 성능이다. KoBERT, KoELECTRA에서는 성능이 되려 떨어진 것을 확인할 수 있었고, 한국어 댓글 데이터로 학습한 KcBERT, KcELECTRA에서는 성능이 향상되었다. 또한, Auto-regressive 모형을 활용한 GPT-2 모델인 KoGPT-2에서도 성능이 향상되었다.

[표 9] CORN 분류 모형의 KOCO-hate 분류 성능

	KoBERT	KcBERT	KoELECTRA	KcELECTRA	KoGPT2
Accuracy	0.55	0.63	0.64	**0.71**	0.60
F1-score	0.54	0.64	0.63	**0.71**	0.59
MAE	0.53	0.39	0.38	**0.30**	0.44
MSE	0.68	0.44	0.42	**0.32**	0.53

4) KOCO-hate Test 분류 성능

KOCO 연구팀에서 개최하고 있는 Kaggle competition에서는 KOCO-hate 데이터셋의 test 데이터셋을 예측한 후, 예측한 레이블을 csv 등의 형태로 변환하여 Kaggle competition에 제출하면 채점 후 F1-score를 확인할 수 있

다. KOCO 연구팀에서 실험한 KoBERT 모형의 성능은 F1-score 0.52였다.

본 연구에서는 각 모델의 기본 분류 모형, CORAL 분류 모형, CORN 분류 모형 중 가장 좋은 성능을 낸 모형을 사용하여 KOCO-hate 데이터셋의 test 데이터를 예측하여 Kaggle에 제출하였다. 표 10은 모델별 KOCO-hate test 데이터에 대한 분류 성능을 나타낸 것이다. 본 연구에서 제안한 순서형 회귀 모형인 CORN을 사용한 KcELECTRA 모델에서 베이스라인 모델의 성능인 F1-score 0.52보다 0.08만큼 성능이 향상되었다.

[표 10] 모델별 KOCO-hate test 데이터 분류 성능

	KoBERT -CORAL	KcBERT -CORN	KoELECTRA -CORAL	KcELECTRA -CORN	KoGPT2 -CORN
F1-score	0.55	0.57	0.56	**0.60**	0.53

VI. 결론 및 향후 연구

컴퓨터 통신 기술의 발달로 온라인 네트워크를 통한 사회활동이 증가하고 있다. COVID-19 바이러스의 여파로 많은 오프라인 활동이 메타버스와 같은 온라인 가상세계에서의 활동으로 전환되면서 온·오프라인의 경계가 모호해졌다. 온라인활동이 늘어나면서 YouTube, Instagram과 같은 SNS 플랫폼을 통한 소통이 활발해졌다. 하지만, 이를 악용하여 혐오 발언 혹은 차별적 발언을 하여 대상에게 정신적인 피해를 주어 극단적인 선택을 하는 경우도 발생하였다. 따라서 이를 사전에 방지할 수 있는 악성댓글 탐지 시스템의 필요성이 대두되었다.

악성댓글을 빠르고 정확하게 탐지하기 위해 여러 방법이 제안되었다. 단어에 악의성 수치를 부여하고 이를 토대로 SVM 기법을 활용한 악성댓글 분류 모델을 제작하여 문장 내 단어의 악의성 수치에 따라 악성댓글을 분류하거나 신조어 사전과 댓글 충격량을 기반으로 악성댓글을 분류하는 방법이 제안되었으나, 모두 데이터베이스가 업데이트되지 않으면 모델의 성능이 떨어진다는 한계점이 있었다.

따라서 본 연구에서는 데이터베이스에 의존하지 않고, 단어 간의 관계성을 바탕으로 악성 댓글을 분류하면서 악성 댓글 분류 성능을 높이고자 하였다. KOCO-hate 데이터셋이 혐오의 정도에 따라 레이블링이 수행된 것에 착안하여 순서형 척도를 분류할 때 좋은 성능을 보이는 순서형 회귀 모형을 사용하여 한국어 자연어처리 모델의 악성댓글 분류 성능을 개선시켰다.

순서형 회귀 모형을 활용한 악성 댓글 분류작업의 성능을 확인하기 위해 사전학습된 한국어 자연어처리 모델에 순서형 회귀 모형인 CORAL 모형과 CORN 모형을 사용하여 기본모형과 성능을 비교하였다. 순서형 회귀 모형을 활용한 모델이 기본모형보다 향상된 성능을 보였으며, 이를 통해 문장의

혐오 정도에 따른 레이블의 순서 정보를 학습하는 것이 성능에 영향을 준다는 것을 알 수 있었다.

특히 CORAL 분류 모형은 importance weights를 조절하여 클래스별 중요도를 설정할 수 있다. 실제 인터넷에서 악성댓글 탐지 모형으로 CORAL 분류 모형을 사용한다면, 해당 사이트에서 허용 가능한 혐오 발언의 정도에 따라 importance weights를 설정함으로써 탐지할 악성댓글의 혐오 정도를 조절할 수 있다. 나아가 나이별로 노출되는 댓글을 조절할 수도 있다.

향후 연구에서는 순서형 회귀 모형을 사용하여 혐오 발언 음성 데이터셋을 분류하고자 한다. CORAL과 CORN은 얼굴 이미지에 따른 나이 예측을 위해 제안된 프레임워크이다. 따라서 후속 연구가 주로 CNN 네트워크에서 이루어졌고, 좋은 성능을 보였다. 음성데이터 분류 연구도 CNN 기반의 모델로 많이 수행되기 때문에 혐오 정도에 따른 순서형 레이블을 가진 혐오 발언 음성데이터 분류 연구에서 순서형 회귀 모형을 사용한다면 성능을 향상시킬 수 있을 것이다. 또한, 최근 SNS에서는 이미지와 텍스트가 결합된 형태의 밈(meme)이 더 많이 사용되고 있다. 악성 댓글과 악성 밈은 모두 정신적인 피해를 줄 수 있는 악성 콘텐츠로 볼 수 있으므로 악성 밈과 정상 밈을 분류할 수 있는 방법론을 탐색할 것이다.

ACKNOWLEDGEMENTS

본 논문은 한국컴퓨터정보학회 학술대회에서 발표한 ‘순서형 회귀분석을 활용한 악성댓글 분류’[24] 논문을 바탕으로 확장하여 후속 연구를 수행한 논문입니다. 본 논문을 지도해주신 박새롬 교수님께 감사드립니다.

참고 문헌

[1] J. MOON, W. CHO, and J. Lee., BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection, Proceedings of 8th international workshop on NLP for social media, 25-31, 2022

[2] "악플: 팬데믹 기간 온라인 혐오 발언 20% 증가", BBC News Korea, November 16, 2021.

[3] 윤지예, "악성 댓글, 지금은 어떤 상황인가.", campusN, July 20, 2020.

[4] Jinju, Hong., Sehan, Kim., Jeawon, Park., and Jaehyun, Choi., A Malicious Comments Detection Technique on the Internet using Sentiment Analysis and SVM, Journal of the Korea Institute of Information and Communication Engineering, 260-267, 2016.

[5] Shin, M., Chin, H., Song, H., Choi, J., Lim, H., and Cha, M., “Hate Speech Detection in Chatbot Data Using KoELECTRA”. In Annual Conference on Human and Language Technology, 518-523, 2021.

[6] Lee, Hyun-Sang, Hee-Jun Lee, and Se-Hwan Oh. “A Study on the Toxic Comments Classification Using CNN Modeling with Highway Network and OOV Process”, Journal of Information System, Vol. 29:3, pp.103-117, 2020.

[8] Krippendorff, Klaus. "Computing Krippendorff's alpha-reliability." (2011).

[8] VASWANI, Ashish, et al. Attention is all you need. Advances in neural information processing systems 30, 2017.

[9] Devlin Jacob, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina., BERT: Pre-training of deep bidirectional transformers for language unders

tanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805., 2018.

[10] SKTBrain KoBERT, KoBERT, GitHub repository. https://github.com/SKTBrain/KoBERT. 2019.

[11] J. Lee., KcBERT: Korean comments BERT, In Proceedings of the 32 nd Annual Conference on Human and Cognitive Language Technology, pages 437 - 440, 2020.

[12] K. Clark, M. T. Luong, Q. V. Le and C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, In International Conference on Learning Representations, 2020.

[13] J. Park., KoELECTRA: Pretrained ELECTRA Model for Korean, Github repository, https://github.com/monologg/KoELECTRA. 2020.

[14] J. Lee. KcELECTRA, Github repository. http://github.com/Beomi/KcELECTRA. 2021.

[15] Radford Alec, Wu Jeffrey, Child Rewon, Luan David, Amodei Dario, and Sutskever Ilya, Language Models Are Unsupervised Multitask Learners. Technical Report. OpenAI, 2019.

[16] Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 2019.

[17] SKT-AI KoGPT2, KoGPT2, Github repository. https://github.com/SKT-AI/KoGPT2. 2021.

[18] Niu, Zhenxing, et al. "Ordinal regression with multiple output cnn for age estimation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[19] Cao, W., Mirjalili, V., Raschka, S.: Rank-consistent ordinal regression

for neural networks. arXiv preprint arXiv:1901.07884, 2019.
[20] Tian, Li, et al. "Learning discriminative representations for fine-grain ed diabetic retinopathy grading." 2021 International Joint Conference on Ne ural Networks (IJCNN). IEEE, 2021.
[21] Han, Wenjing, et al. "Ordinal learning for emotion recognition in cust omer service calls." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.
[22] Shi, X., Cao, W., Raschka, S.: Deep neural networks for rank-consis tent ordinal regression based on conditional probabilities. arXiv preprint ar Xiv:2111.08851. 2021.
[23] Y, Ha., J, Cheon., I, Wang., M, Park., and G, Woo., A Filtering Meth od of Malicious Comments Through Morpheme Analysis, JOURNAL OF THE KOREA CONTENTS ASSOCIATION, 750-761, 2021.
[24] Lee, Seyoung, and Saerom Park. "Hate Speech Classification Using O rdinal Regression." Proceedings of the Korean Society of Computer Inform ation Conference. Korean Society of Computer Information, 2021.

ABSTRACT

A Study on the Performance Improvement of Artificial Intelligence HateSpeech Detection Model Using Ordinal Regression Model

Seyoung Lee
Department of Future Convergence Technology Engineering
Graduate School of
Sungshin University

Online activities have become more active in the wake of the development of computer communication technology and the COVID-19 virus. In particular, as SNS activities that allow users to enjoy online content such as YouTube and Tiktok increase rapidly, they often enjoy content on SNS platforms and express their opinions through online comments. Because anonymity is guaranteed due to the nature of the online, malicious comments containing hate speech or prejudice remarks are easily written by exploiting freedom of expression. Online malicious comments cause mental damage to objects that exist offline. Since malicious comments can lead to extreme choices, preventive measures and regulatory measures for malicious comments are urgently needed.

Among the KOCO (KOrean COmments) datasets that collected

Korean comment data to learn the Korean malicious comment classification model, the KOCO-hate dataset labeled malicious comments with normal, aggressive, and severe hate speech depending on the degree of disgust. Therefore, since malicious comment classification is a multi-classification problem according to the degree of hate speech, we propose a malicious comment classification model using an ordered regression model that is effective in classifying ordered classes to utilize ordered information of each class. First, for hate speech classification, the CORAL (COnsistent Rank Logits) framework and CORN (Conditional Ordinal Regression for Neural Network) framework were applied to the pre-learned Korean natural language processing model.

When comparing the classification performance of the basic model, the CORAL model, and the CORN model, it was confirmed that the performance was improved in the CORAL and CORN models using the ordinal regression model.