DistilBERT 를 이용한 개인 맞춤형 악플 학습 모델 고안

박건우, 박태현, 권민혜
숭실대학교
{parkgw0013, 20201588}@soongsil.ac.kr, minhae@ssu.ac.kr

Designing a Personalized Comment Moderation Model Using DistilBERT

Geonwoo Park, Taehyun Park, Minhae Kwon
Soongsil University

요 약

본 논문은 DistilBERT 기반의 개인 맞춤형 악플 학습 모델을 제안한다. 10,000 개의 한국어 댓글 데이터셋을 활용하여 모델을 학습시켰으며, 사용자의 주관적 판단을 통해 지속적으로 학습 데이터를 업데이트 하는 기능을 갖추었다. Softmax 함수를 활용하여 모델의 결과를 정확하고 투명하게 수치화함으로써, 사용자는 모델의 판단을 더 신뢰할 수 있다. 본 연구는 SNS 시스템에 쉽게 통합될 수 있으며, 비속어가 포함되어 있지 않은 악플 판단의 정확성을 높이는 데 기여한다.

Ⅰ. 서론

SNS(Social-Network-Service)와 온라인 커뮤니티는 다양한 사람들이 자유롭게 자신의 생각을 표현할 수 있는 공간으로 자리매김하였다. 그러나 이러한 공간에서 발생하는 악성 댓글들은 개인에게 정서적 고통을 유발할 수 있는 대표적인 문제 중 하나로 꼽힌다. 이런 댓글들에 대비하여 여러 딥러닝 및 NLP 모델이 개발되었지만, 악플을 판단하는 과정에서 욕설이 포함되어 있지 않은 댓글은 탐지하기 어려운 경우가 많다. 인간의 주관적인 판단이 필수불가결한 경우가 이러한 욕설이 포함되어 있지 않고 비꼬는 듯한 부정적인 뉘앙스를 풍기는 표현의 댓글이다.

본 연구는 이러한 한계를 해결하고자 DistilBERT[1]를 기반으로 개인 맞춤형 악플 학습 모델을 제안한다. 10,000 개의 댓글을 인간이 직접 악플과 악플이 아닌 댓글로 구분한 데이터셋을 이용해 모델을 학습시켰다. 본 모델은 댓글을 입력할 때마다 해당 댓글이 악플인지 아닌지 판단한다. 추가적으로, 잘못 분류된 경우에는 사용자의 주관적 판단으로 Feedback 하여 데이터셋에 추가하는 기능도 포함한다. 이를 통해 모델은 점진적으로 개인의 기준에 맞춰 최적화되어 가며, 악플 탐지 성능을 향상시킨다.

Ⅱ. 본론

Ⅱ-i. DistilBERT 모델 선택 이유

DistilBERT 는 BERT 모델의 경량화된 버전이다. BERT 모델을 단순화시켜 학습 및 추론 시간을 크게 줄이면서도 성능은 비슷하게 유지하도록 설계되었다. 여러 NLP 모델(KoBERT, KcBERT), 그리고 순환신경망(RNN) 기반의 Attention Bi-LSTM 등이 있지만, Colab 의 T4 GPU 를 사용 중인 환경에서조차 Runtime 이 길어지는 현상이 발생하므로, 짧은 시간 내로 run 이 완료될 수 있는 DistilBERT 모델을 선택하였다. 본 연구에서 고안한 코드의 전체 Runtime 은 약 8 분 정도 소요되었다. 아래 표.1 은 앞서 말한 모델들의 성능을 연구에서 사용한 데이터셋으로 평가하였을 때 도출된 결과이다.

Model	Accuracy(%)
KcBert	90.6
KoBert	88.2
Attention Bi-LSTM	85.8

표.1 모델 성능 비교

Transformer 계열의 KcBERT 가 가장 높은 정확성을 보였고, RNN 기반의 Attention Bi-LSTM 이 가장 낮은 정확도를 보였다. 본 연구에서 사용한 DistilBERT 의 Accuracy 는 최적의 fine-tuning 이후 최종적으로 82.15%로 측정되었다. KcBERT 는 DistilBERT 보다 2 배 많은 레이어를 가지고 있고, 1.7 배 많은 파라미터를 가지고 있다. 그럼에도 불구하고 DistilBERT 는 정확도 면에서 큰 차이를 보이지 않았고, 연구에 사용한 데이터의 양이 많지 않으므로 DistilBERT 를 사용하는 것이 적합하다고 판단하였다.

Ⅱ-ii. 시스템의 전체적인 구조

본 논문에서 개발한 시스템의 전체적인 구조는 아래 그림.1 에 나타내었다.

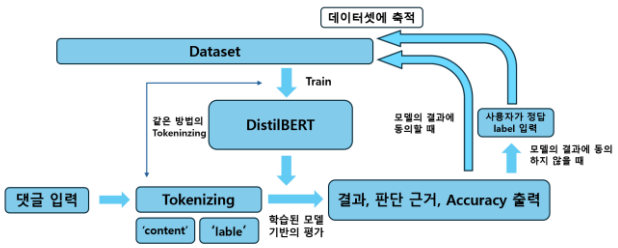

그림.1 시스템의 구조

Ⅱ-iii. Develop

-Dataset

본 연구에서는 10,000 개의 한국어 댓글을 전문가가 직접 악성 댓글과 악성 댓글이 아닌 것으로 구분한 데이터셋을 이용하였다. 이 데이터셋은 Github 에서 공개적으로 접근이 가능하며, 각 댓글은 오른쪽 공백에 악플 여부를 나타내는 레이블로 주석이 달려있다. 정상 댓글은 1, 악플은 0 으로 기재되어 있다.

-PreProcessing&Tokenizing

데이터는 Pandas 라이브러리를 사용해 csv 파일 형태로 로드하고, 각 댓글은 소문자로 변환, 특수 문자 제거, 공백 제거 등의 전처리 과정을 거친다. 다음으로, Transfomer 라이브러리의 DistilBertTokenizer 를 이용해 전처리된 텍스트를 토큰화한다. BERT 계열의 모델은 텍스트 대신 숫자를 처리하기 때문에[2], 텍스트를 토큰화하는 작업은 필수적이다. 모든 입력은 max_length=64 로 padding 하며, max_length 보다 길 경우 Truncation 옵션을 사용하여 텍스트를 자동으로 자르는 과정을 추가한다.

아래 그림.2 는 전체 알고리즘의 Pseudo code 를 나타낸다.

Pseudo Code for Total Algorithm

INITIALIZE DistilBERT model with pre-trained parameters

LOAD dataset from **CSV** file

PREPROCESS data by converting to lowercase, removing special characters, and trimming spaces

SET tokenizer **TO** DistilBertTokenizer

FOR each comment **IN** datatset

  TOKENIZE and **TRUNCATE** comment **USING** tokenizer **WITH** max_length=64

  ADD tokenized comment **TO** processed_data

END FOR

SPLIT dataset **INTO** training_data(80%) **AND** validation data(20%)

TRAIN model **ON** training_data **WITH** parameters

(batch size=64, epochs=10, learning rate=1e-5)

EVALUATE model on validation_data

WHILE true

  INPUT comment

  IF comment **IS** "exit"

    BREAK

  END IF

  PREPROCESS and **TOKENIZE** comment

  PREDICT sentiment using model

  DISPLAY prediction and probabilities

  INPUT user_feedback

  UPDATE dataset with comment and user_feedback

  SAVE updated dataset to **CSV**

END WHILE

그림.2 Pseudo Code for Total Algorithm

-학습 및 평가

Train data set 와 Test data set 을 각각 80%, 20%의 비율로 조정하였다. 분할한 데이터는 PyTorch 의 DataLoader 를 통해 미니 배치로 구성하여 모델 학습에 활용하였다. Hyperparameter 의 경우, 여러 fine-tuning 시도 결과 Batch size=64, epoch=10, Learning rate=1e-5, optimizer=AdamW 로 하였을 때, 최종적으로 가장 높은 accuracy=82.15%를 보였다.

- 판단 근거 출력 및 최종 테스트

모델의 학습과 평가가 끝난 후, 사용자는 댓글을 입력하여 모델을 시험해 볼 수 있다. 추가된 댓글을 데이터셋에 축적해야 하므로 pd.read_csv()를 이용해서 학습했던 dataset.csv 를 불러온다. 그리고 입력한 댓글의 텍스트 전처리 및 토큰화 과정은 학습 전 토큰화 과정과 완전히 동일하다.

판단 근거를 출력할 때, 본래 모델은 각 클래스에 대한 로짓($logit$)값을 출력한다. 로짓 값은 해당 클래스에 속할 로우 스코어($raw\ score$)를 나타낸다. 로짓은 직접적인 확률 값이 아니라, 모델이 계산한 각 클래스의 가능성을 나타내는 점수이다.

이를 정규화하기 위하여 $softmax$ 함수를 사용한다. $softmax$ 함수는 각 $logit$ 에 대해 e^{logit} 을 적용하여 모든 값을 양수로 만드는 함수이다. 이렇게 계산된 값들은 모든 클래스에 대해 합쳐진 후, 각 클래스의 값을 총합으로 나눔으로써 확률로 변환된다. 이러한 과정을 통해 각 클래스의 확률은 0 과 1 사이의 값으로 정규화되며, 두 클래스 확률의 합은 정확히 1이 된다.

실제 댓글 입력을 하였을 때, 예측과 부합한 4 가지 예시와 예측과 부합하지 않은 2 가지 예시를 아래 표.2 에 나타내었다.

댓글	분류 결과	악플	정상 댓글
안녕하세요	정상 댓글	0.00	1.00
좀 씻고 다녀라	악플	0.99	0.01
너무 귀여운데?	정상 댓글	0.01	0.99
멍청하네	악플	0.98	0.02
너는 노력해도 안 돼	정상 댓글	0.43	0.57
ㅇㅇ	악플	0.77	0.23

표.2 실제 시스템 테스트 결과

표.2 에서 볼 수 있듯이, "안녕하세요"와 같이 위 4 개의 예시는 일반적인 예상과 비슷하게 출력되었음을 알 수 있다. "좀 씻고 다녀라"도 비속어나 욕이 포함되어 있지 않음에도, 0.99 의 확률로 일반적인 예상과 같이 악플로 출력된 것을 볼 수 있다. 하지만 아래 2 가지 예시는 일반적인 문맥상 표.2 에 출력된 결과와 반대의 결과를 예측하는 경우가 많다. 데이터셋을 확인해 본 결과, '악플'인 경우 "ㅇㅇ"이 포함되어 있는 경우가 많았고, '정상 댓글'인 경우 "노력"이라는 단어가 포함되어 있는 경우가 많았다. 여러 번의 데이터셋 축적 끝에, 두 댓글을 서로 반대의 분류 결과로 출력하는 데 성공하였다.

Ⅲ. 결론

본 연구에서는 DistilBERT 를 기반으로 한 개인 맞춤형 악플 학습 모델을 개발하였다. BERT 의 경량화 버전인 DistilBERT 를 사용했음에도, 짧은 시간 내에 정확도 82.15%라는 높은 결과를 얻을 수 있었다.

10,000 개의 한국어 댓글 데이터셋을 사용하여 사전 훈련된 모델을 미세 조정하고, 사용자의 주관적 판단을 통해 지속적으로 학습 데이터를 업데이트하는 기능을 갖추었다. 욕이나 비속어가 없을지라도, 악플로 주관적으로 판단하여 데이터셋에 축적하는 기능이 있다는 점에 의의가 있다. 욕이나 비속어를 포함한 경우에도, 정상 댓글인 경우 주관적으로 판단하여 올바르게 분류할 수 있다. 또한, $softmax$ 함수를 추가함으로써, 모델이 단순히 결과만을 전달하는 것에 그치지 않고 결과에 대한 신뢰성과 정확성을 수치적으로 표현할 수 있게 한다. 이러한 확률 값을 제공함으로써 모델의 투명성을 높이며, 사용자는 모델의 판단을 더 신뢰할 수 있다.

결론적으로 본 연구에서 개발한 알고리즘은 인간의 주관적 판단과 딥러닝 기술을 결합함으로써, 사용할수록 개인에게 맞춤화되는 Interactive Feedback System 이다. 특정 단어가 탐지되지 않으면, 현재 기술로서는 악플에 대한 정확한 판단을 하는 것이 어렵다[3]. 조롱하는 댓글과 비꼬는 듯한 뉘앙스의 댓글을 더 정확하게 분석할 수 있는 알고리즘 개발이 필요하다고 생각하여 본 연구를 진행하였다. 해당 모델은 사용할수록 성능이 강화될 것이며, 현재 SNS 시스템에 부분적인 적용이 가능하다.

참 고 문 헌

[1] 신동환, 김한석, 이수진. "데이터 전처리를 간소화한 자연어처리 기반의 악성코드 탐지모델," 한국정보기술학회논문지, vol. 22, no. 2, pp. 231-232, 2, Feb, 2024

[2] 노동훈, 민재옥, 우소연. "특허상담 자동분류의 성능 향상 방안 연구: 트랜스포머 기반 인공지능 모델 버트(BERT)를 활용," 지식재산연구, vol. 19, no. 1, pp. 159-177, Mar, 2024

[3] 이신행. "편향적 인공지능: 네이버의 악플 탐지용 인공지능 '클린봇'이 판별한 혐오표현의 유형 분석," 사이버커뮤니케이션학보, vol. 38, no. 4, pp. 33-75, Dec, 2021