욕설 검열 인공지능 모델 개발 탐구

20928 정준영

내용

I. 서론

1학년때 자율 활동으로 '온라인 스트리밍 채팅 내 이모트가 포함된 혐오 표현 탐지: 딥러닝 모델을 기반으로 (김재헌, 원동희, 차미영)'을 읽고 탐구를 진행하여 세특[1]이 작성되었다. 이후 해당 논문에서 사용된 LSTM에 대해 관심이 생겼고 이를 이용하면 전범위에 대하여 여러 욕설과 비속어 등의 표현을 탐지 할 수 있으리라 생각하여 본 탐구를 진행하게 되었다.

처음 시작은 언어의 앞뒤 문맥을 파악하여 욕설을 탐지해도록 순환 신경망(RNN)의 일종인 '양방향 장단기 기억 신경망(Bi-LSTM)'을 사용하여 모델을 학습시키고자 하였으나, 추가적인 조사와 연구를 통하여 구글에서 개발한 '트랜스포머(Transformer)' 이론과 'Bert' 모델에 대해 알게 되었고, 이 두 모델을 비교하여 최종적으로 초기 계획과는 다르게 성능과 정확도가 더 우수한 Bert모델을 사용하여 학습을 진행하였다.

본 탐구 보고서에서는 탐구 진행 과정을 그 흐름대로 설명하며 어떠한 생각을 통해 어떤 방법으로 어떠한 결과를 도출해 내었는지를 중심으로 설명할 것이다. 데이터 수집부터 데이터 전처리와 분석, 모델 학습과 검증 등 탐구 과정 중 매우 많은 파이썬 코드 파일들을 작성하였으며 본 탐구의 핵심인 모델 학습코드 외에는 따로 언급하지 않겠다.

---

[1] 혐오 표현 탐지 인공지능과 회피하려는 사람들의 모습 속에서 특정 스트리밍 사이트의 혐오 채팅 데이터를 분석을 통해 혐오 표현 이모지와 스트리머의 인종을 비교하여 새로운 혐호 표현 검출 모형을 만들 수 있음을 파악하고 이에 관한 추가 탐구를 계획하는 모습을 보여줌. (자율 세특 내용 中)

II. 데이터 수집

우선 욕설 탐지를 위해서는 모델 학습을 진행하여야 하는데, LSTM과 Bert 모두 지도학습[2]에 해당되므로 학습 데이터와 레이블(정답) 데이터가 필요하다. 모델의 목적은 채팅(댓글)속 욕설의 탐지이므로 학습 데이터 또한 댓글을 통하여 수집하였다. 욕설 뿐 아니라 다양한 비속어와 은어 등 부적절한 표현들을 골구루 수집하기 위해 대표적인 뉴스 플랫폼인 네이버뉴스[3](이하 뉴스)와 대표적인 커뮤니티인 디시인사이드[4](이하 디시)에서 수집하였다. 댓글 데이터를 수집하기 위해 웹 크롤링을 사용하여 직접 해당 사이트에서 댓글들을 긁어왔다. 처음에는 뉴스와 디시에서 댓글이 편향적일까 봐 걱정이 되었지만 생각외로 뉴스뿐만 아니라 디시에서도 다양한 정치스펙트럼이 나타나 우려했던 걱정은 안해도 되었다.

뉴스에서는 '언론사별 댓글 많은 순 랭킹'에서 24-07-01에서 24-07-31까지 날짜를 범위로 각 언론사별 1~5위 뉴스의 댓글들을 수집하였다. 디시에서는 인기 게시글을 모아둔 카테고리[5]인 '실시간 베스트 갤러리'에서 24년 7월 중 여러 번의 수집으로 총 443개의 게시글에서 댓글을 수집하였다. 이렇게 수집하여 중복된 댓글과 너무 짧은 댓글 등 데이터로써 가치가 낮은 댓글들을 삭제하는 처리 과정 후 최종적으로 남은 댓글은 뉴스에서 2,084,812개, 디시에서 1,864,759개로 총 3,949,571개의 댓글을 수집하였다. Txt파일의 용량이 205,298,026byte 이므로 한글당 3byte로 환산하면 최소 글자수가 68,432,675, 댓글당 평균 글자수가 17개이다.

---

[2] 입력 데이터와 정답 레이블이 함께 제공되는 학습 방법

[3] 한국 트래픽 전체 7위(2024-05-01, SimilarWeb)

[4] 한국 트래픽 전체 5위. 1위: 네이버 / 2위: 구글 / 3위: 유튜브 / 4위: 다음 / 6위: 나무위키

[5] 추후 알게 된 사실에 따르면 자동으로 인기 게시글이 올라오는 시스템이 아닌 '알바(운영자)'가 직접 선정하여 게시글을 올리는 방식이라 실베에 올라왔다 해서 모두 다 인기가 있고 댓글이 많은 것은 아니라고 한다 (실시간 베스트 갤러리, 나무위키)

III. 데이터 가공 및 분석

데이터를 수집하였으니 해당 데이터들에 대한 레이블을 달아줘야 한다. 그러나 댓글이 300만개가 넘는 상태에서 하나하나 수작업으로 다는 것은 불가능하다 판단하여 효율적인 라벨링을 위해 댓글에서 욕설 등의 부적절한 단어를 추출하여 해당 단어가 포함되었는가를 기준으로 욕설/정상 댓글을 분류하도록 만들었다.

댓글에서 부적절 단어를 추출하기 위해 뉴스와 디시 댓글에서 많이 사용되는 단어를 우선 뽑아보았다. 해당 단어들은 뉴스는 'news_freq_word.db', 디시는 'freq_word.db' 그리고 이를 정리 및 축소하여 최종적으로 'high_freq_words.txt'에 저장해 두었다. 이후 네이버 뉴스에서는 욕설을 차단하는 '클린봇' 기능으로 댓글에 직접적인 욕설이 매우 적다는 점을 이용하여 디시 댓글에서는 많이 사용되지만 뉴스 댓글에서는 적게 사용되는 단어를 추출하여 'high_freq_dc_words.json'에 각 단어의 출현 횟수를 포함하여 저장하고, 해당 단어들을 수작업으로 욕설 여부를 라벨링하여 1487개의 욕설 리스트를 만들었다. 이후 추가적으로 빠져있는 욕설 및 비속어에 대해 수작업으로 추가해 주었다.

뉴스와 디시의 댓글에서 출현 빈도가 높은 단어들을 워드클라우드(Word Cloud)를 통해 표현해봤다. 좌측이 뉴스, 우측이 디시이다.

해당 사진에서 볼 수 있듯 디시의 댓글에서 욕설이 훨씬 많이 보이는 것을 확인할 수 있다.

IV. BiLSTM과 Bert

두 모델 다 딥러닝 모델로, 전자는 순환 신경망(RNN)의 일종이며 후자는 자연어 처리(NLP)의 일종이다.

BiLSTM은 기존 LSTM의 업그레이드 버전이다. LSTM은 학습이 계속될수록 초기에 학습된 내용을 잊어버린다는 RNN의 단점을 보완하기 위해 학습을 할 때 이전 과거 학습 데이터까지 계속 함께 제공하여 학습의 정확도를 높이는 아키텍쳐이다. 여기서 과거 데이터 뿐 아니라 미래데이터까지 함께 제공받으면 BILSTM이 된다. 앙방향 LSMT은 한방향으로만 학습하는 것이 아닌 양방향으로 학습을 진행한다. 즉 학습방향을 서로 다르게 총 두번 LSTM계층을 지나게 되는거다. 이를 통해 시계열 데이터에서 앞과 뒤의 '문맥'을 파악할 수 있게된다. 예를들어 *"대건고 OOO 미술 교사는 수업을 항상 10분 늦게 마칠 정도로 매우 열정적이시다. 덕분에 종례가 늦어져서 학원에 지각할 뻔했다."* 라는 댓글이 있고 해당 댓글이 칭찬인지 비꼬는 것 인지를 판단한다 했을 때, RNN의 경우 해당 댓글에 욕설이 존재하지 않고 종례가 늦은것과 10분 늦게 마친것에 연관성을 기억하지 못하므로 **칭찬**, LSTM은 과거 정보만을 제공받으므로 10분 늦게 마쳤고 지각할 뻔 했지만 결국 안했으므로 **칭찬**, Bi-LSTM은 역방향에서 '학원에 지각할 뻔 했는데 그 이유가 미술 선생님이 10분 늦게 마쳐줬으므로' **비꼼** 이라 판단 할 수 있는 것이다.

Bert는 트랜스포머 기반 모델로 Bi-LSTM과 같이 양방향으로 처리하여 문맥 파악을 잘 한다. Bert는 미리 훈련된 모델로 해당 모델의 미리 훈련한 데이터에 학습시키고자 하는 데이터를 추가 학습시켜 더 정확한 모델을 개발 할 수 있다. 트랜스포머는 거대 언어 모델 LLM의 시초가 되는 이론으로 이를 이용하여 GPT 등 다양한 언어 모델이 탄생하였다. 기존 모델과는 다르게 어텐션(Attention)이라는 매커니즘을 도입하여 병렬 계산을 통하여 학습 속도를 획기적으로 단축시켰다. RNN의 순차적 학습, LSTM의 기억 학습, Bi-LSTM의 양방향 학습 등을 모두 효율적으로 합치며 속도는 줄여 효율적이며 매우 정확한 획기적인 모델이다. 여기서 어텐션이란 영단어 뜻 그대로 '집중'이라는 뜻이다. 즉 중요한 핵심에 집중하여 컴퓨터가 문장을 이해할 때 어떠한 단어에 포커

스를 맞출지를 판단한다. 예를들어 *"OOO 수학 선생님은 대건고에서 근무하신다.", "OOO 선생님은 2학년 담임이시다.", "OOO 선생님은 미적분을 잘 가르치신다."* 라는 댓글들에서 컴퓨터가 OOO에 '집중'한다 하면 컴퓨터는 OOO과 수학, OOO과 선생님, OOO과 대건고••• 이렇게 '집중' 단어와 다른 단어와의 관계를 파악하여 최종적으로 컴퓨터는 *'OOO 선생님은 대건고 수학 교사이며 2학년 담임을 담당하고 미적분을 잘 가르치신다*라 생각하게 된다. 이후 여기서 *"OOO선생님은 담당 과목이 XXX이다."* 라는 댓글을 보았을때 XXX는 '미적분' 이라는 추론 또한 할 수 있게 된다. 본 탐구에서는 한국어 댓글을 다루고 있으므로 한국어에 특화된 KoBert 모델을 사용하였다

.

V. 모델 학습

BiLSTM과 Bert 모델 모두 코드를 파이썬으로 작성하여 학습을 진행해 보았다. 우선 두 모델의 성능을 비교하기 위해 랜덤으로 정렬한 댓글 데이터 3만개만을 사용하여 학습을 진행했다. 학습 진행 후 모델 평가에서 BiLSTM의 정확도는 약50%대(결과 기록 데이터를 잃어버려 정확한 수치 알 수 없음), Bert의 정확도는 79.59%가 나왔다. 이후 BiLSTM에서의 Epoch(학습 반복 횟수)를 5로 Bert의 5배로 늘렸으나 여전히 58%대의 정확도를 보여 Bert가 BiLSTM보다 욕설 탐지 성능이 우수하다 판단하였다.

이후 Bert로만 학습을 진행하였다. 한번에 하지 않고 학습을 나누어 진행하였다. 학습을 할 때는 기존의 학습된 모델을 다시 불러와 추가 학습을 시켰다. 다음은 각 모델별 학습 데이터 수와 정확도이다.

- 모델 1: 30,000개 / 79.59%
- 모델 2: 200,000개 / 86.72%
- 모델 3: 220,000개 / 91.20%
- 모델 4: 400,000개 / 93.26%
- 모델 5: 700500개 / 95.31%

총 1,550,500개의 데이터를 학습하여 95.31%의 정확도를 가진 모델을 얻었다. 학습 데이터는 댓글의 글자수를 기준으로 적절하게 필터링하여 랜덤으로 정렬하여 선별하였다. 뒤로 갈수록 더 길거나 짧은 댓글 데이터도 추가하여 다양한 길이의 댓글에서도 욕설을 탐지할 수 있도록 만들었다. 처음에는 수집한 데이터 전부(3,949,571개)를 학습시키고자 하였으나 학습 예상 시간이 240시간(10일)을 넘어가 이는 무리라고 판단하여 1,550,500개만 학습하였다.

VI. 결론

Bert 모델을 이용하여 욕설을 비롯한 비속어, 은어, 비하 표현 등 여러 어린이들에게 부적절한 단어들을 탐지해내는 인공지능 모델을 개발해 보았다. 이를 위해 직접 뉴스와 커뮤니티 사이트의 댓글들을 스크랩핑하여 이를 가공하고, 데이터를 분석하여 욕설 단어 리스트를 쉽게 추출해내어 전부 다 수작업으로 하지 않고 자동으로 라벨링을 해주어 시간과 비용, 에너지를 절약하였다. 이후 BiLSTM과 Bert를 비교해보고 더 목적에 적합한 모델을 선정하여 정확도가 높은 모델을 추구하였고, 추가적인 학습으로 모델의 정확도를 95%대까지 끌어올렸다. 남은 데이터까지 포함하여 추가적인 학습을 진행한다면 정확도를 97%대까지 올릴 수 있을것으로 예상된다.

대체로 학습은 우수하게 된것으로 보이나 일부 허점이 보여 아쉽다. 예를들어 1(욕설)로 라벨링된 댓글 대부분에는 정치인의 이름이 들어있는 경우가 대부분이라 모델이 정치인의 이름을 욕설로 인식한다. 이는 데이터를 더 가공하고 모델을 수정하면 보완 될 것이라 생각된다.

본 탐구를 7월초부터 시작하여 꾸준하게 진행하며 여러 변수들과 문제점들이 많았다. 그럴때마다 문제를 해결하기 위해 많은 시간을 쏟고 때로는 하나의 문제에 2~3일동안 하루 종일 붙잡고 있으면서 많이 힘들었으나 결과물이 기대보다 더 우수하게 나와 뿌듯함과 성취감을 느꼈다.

3학년때는 본 탐구에 이어 '댓글 데이터 라벨링 매커니즘 정교화, 필터별 단어 리스트 생성, 인공지능 모델 추가 학습 및 정교화, 필터 및 모델 기반 욕설 탐지 API 구축, API를 이용한 웹 페이지 상의 욕설을 탐지해내어 자동으로 검열해주는 브라우저 확장프로그램 개발'를 수행하여 *'필터 및 인공지능 기반 댓글 내 욕설 탐지 API 구축 및 확장프로그램 개발'* 를 탐구 할 것이다.

해당 탐구에서 사용된 코드들은 깃허브에 업로드 되어있다. 또한 progress_img 폴더내에 수시로 탐구 과정을 스크린샷 한 기록이 남아있다. 이외 데이터들과 모델은 용량이 너무 커서 구글 드라이브에 업로드 해두었다. 전체 용량은 3.95GB이다.

VII. 출처 및 참고문헌

https://www.ibm.com/kr-ko/topics/recurrent-neural-networks

https://kr.mathworks.com/discovery/lstm.html

https://velog.io/@pjimin0309/%ED%8A%B8%EB%9E%9C%EC%8A%A4%ED%8F%AC%EB%A8%B8-%EA%B0%9C%EB%85%90

https://blog.naver.com/young_treasure/223100075171

https://namu.wiki/w/BERT

https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf(번역기로 번역하여 참고함)