감성 분석 기반의 시 추천 웹 서비스

A Poem Recommendation Web Service Based on Sentiment Analysis

요 약

본 논문은 네이버 영화 리뷰 데이터(NSMC)를 학습시킨 koBERT 언어 모델을 활용하여 사용자가 입력한 자신의 하루에 대한 감상평을 감성 분석하고, 이와 적합한 현대시를 제공하는 웹 서비스를 제안한다. 극성을 가진 NSMC 데이터를 변형해 모델을 학습한 결과, 89%의 정확도를 보였다. 제공된 감성 분석에 대해 88%, 현대시에 대해 72%의 만족도를 보였고, 원인과 결과에 따라 같은 감성이라도 다른 감성으로 해석될 경우를 제외한다면 잘못 분류되어 제공된 시더라도 다양하게 해석될 수 있는 시의 특징에 따라 다소의 오류가 있어도 사용자가 이를 자신의 상황에 맞게 해석할 수 있음을 확인했다.

1. 서 론

감성 분석에 대한 관심[1]이 높아지며, 주어진 데이터셋으로 다양한 모델을 사용해 감성 분석의 성능을 높이는 연구가 활발히 이루어지고 있다[2]. 이러한 감성 분석 연구는 2000년대 이후 본격적으로 확대되어 영화평, 뉴스 기사 등의 데이터에 적용하여 선호도 분석 및 챗봇의 자연어 생성 등의 응용에서 사용되고 있다[3]. 이러한 연구에 요구되는 데이터셋은 일반적으로 1,000~10,000개 이상으로 구성되며, 목적에 적합한 데이터셋이 없다면 크롤링 또는 타플랫폼에서의 서베이 등을 통해 새롭게 구축해야 한다[4].

의미있는 데이터셋 구축에는 긴 시간이 소요되며, 구축된 데이터셋을 바탕으로 제공되는 서비스 역시 제한될 수 밖에 없다. 본 논문에서는 일반에게 공개되어 현재 다양한 용도로 사용되고 있는 네이버 영화 리뷰 데이터(NSMC; Naver Seniment Movie Corpus)[5]를 활용하고, 사전 학습시킨 감성 모델을 효과적으로 이용하여 사용자가 입력한 '자신의 하루'에 대한 감상평과 적합한 현대시를 제공하는 웹 서비스를 제안한다.

2. 관련 연구

2.1 KoBERT

KoBERT는 SKT의 T-Brain에서 개발한 모델로서, 기존의 구글에서 개발한 BERT[6] 언어 모델의 한국어 성능 한계를 극복하기 위해 만들어졌다[7]. 한글 위키 기반으로 수집한 한국어 문장을 이용하여 대규모 말뭉치를 학습하였고, 한국어의 불규칙한 언어 변화의 특성을 반영하였다[8]. 또한, PyTorch와 Tensorflow와 같은 다양한 딥러닝 프레임워크와 API를 지원한다.

3. 연구 방법

3.1 데이터 셋

본 논문에서는 사용자가 입력한 문장의 감성 분석을 위해 Naver Sentiment Movie Corpus(NSMC) 데이터셋을 사용한다. 본 데이터는 영화에 대한 리뷰를 긍정, 부정으로 레이블링한 것으로, id, document, label 총 3개의 열로 구분되어 있고, 여기서 document는 각각 140자 이내의 길이이므로 사용자가 입력할 한 문장에서 세 문장 가량의 데이터 형태와 유사하다. 또한, NSMC 데이터는 긍정, 부정으로 레이블링 되어 있으므로, 이를 바탕으로 학습시킨 모델 아키텍처를 이용해 사용자가 입력한 문장의 감정을 긍정, 부정으로 나누어 적절한 시 데이터를 쉽게 제공할 수 있다.

본 연구에서는 먼저 사용자에게 입력받은 문장의 감성을 분석한 후, 각 감정에 맞게 분류된 시 데이터를 제공한다. 이를 위해 트위터 '현대시봇'의 트윗 데이터를 2021년 4월 1일을 기준으로 크롤링하였고, 총 3,000개의 현대시 데이터를 수집하였다. 해당 데이터는 줄바꿈 기호(엔터)로 행 구분이 되어 있으며, 연단위로 나누어 업로드했기 때문에 전처리하기 쉽고 한정되어 있는 1900년대의 시 데이터를 다량으로 수집할 수 있다.

3.2 koBERT를 이용한 감성 분석 모델 생성 방법

본 논문에서 제안하는 시스템은 그림 1과 같이 크게 3단계로 이루어진다. 1단계 과정인 모델 아키텍쳐 생성은 본 연구의 핵심 단계이며, 모델 생성이 완료되어야 다음 단계를 진행할 수 있다.

먼저, NSMC 데이터셋으로부터 document와 label을 추출 및 변형시켜 koBERT에 학습시킨다. 학습 이전에 koBERT에 맞는 입력값으로 해당 데이터를 변환시켜야 하므로, pretrained한 koBERT Tokenizer를 임포트하여

15만개의 NSMC 데이터의 document 열에 해당하는 데이터를 [token input, segment input, mask input] 형태로 변환시킨다. 여기서 token은 문장 토큰, mask는 토큰화된 문장에서 패딩이 아닌 부분은 1, 패딩 부분은 0으로 표시한 것을 나타내며 segment는 문장의 전후관계를 구분해주는 것을 의미한다. NSMC의 label 데이터는 별도의 label 변수를 선언하여 각각의 값을 저장해준다.

각각의 token, mask, segment 입력값과 label 데이터는 document와 label을 열로 하는 numpy array로 저장해 이를 koBERT의 train data로 사용하며, RAdam Optimizer를 이용하여 학습 안전성을 높인다[9].

학습시킨 NSMC 데이터가 긍정, 부정으로만 레이블링 되어 있으므로 이를 바탕으로 생성된 감성 모델은 구분할 수 있는 감성의 한계가 있다. 따라서 감성모델을 보다 확장하기 위해 가지고 있는 도메인 데이터와 pretrained한 model에 dense layer를 추가하여 새로운 레이어를 쌓아 재학습 시키는 방법을 적용한다. 마지막으로 epochs 수를 늘려감에 따라 적절한 모델을 찾아 해당 모델의 가중치를 저장한다. 가중치를 저장하는 이유는 Python 기반의 마이크로 웹 프레임워크인 Flask 기반의 웹사이트에서 모델을 이용할 때, 모델 아키텍처만 불러와 모델 로드가 가능하게끔 하기 위해서이다.

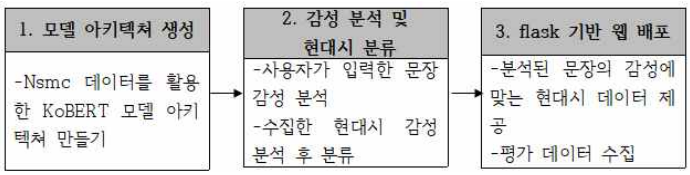

그림 1. 연구 개요 3단계

3.3 감성 분석 및 현대시 분류 방법

사용자가 Flask를 기반으로 한 웹을 통해 데이터를 입력하면, 저장해두었던 모델의 가중치를 불러와 모델을 로드하고 입력한 데이터의 predict score(예측값)를 받아온다. predict score는 0~1 사이의 값이 출력되며, predict score에 따른 감성 분석 기준은 표1과 같이 임의의 임계치에 따라 5가지로 정의한다.

Python의 BeautifulSoup package를 이용하여 트위터 '현대시봇'의 '트윗 및 답글'란의 타겟 URL을 읽어 HTML을 쉽게 검색할 수 있는 상태로 만든다. 현대시 데이터가 속해 있는 태그를 찾아 해당 텍스트값을 가져오고, 불필요한 불용어(stop word)를 전처리 과정을 통해 제거한다. 위에서 정의한 5가지 감성 분석 기준에 따라 현대시의 감성을 분류하고, 사전에 정의한 5가지의 감성 라이브러리 중 해당하는 곳에 저장한다. 이처럼 분류된 현대시 데이터는 데이터베이스에 저장되어, 사용자의 입력값에 대한 감성 분석을 마치고 웹페이지에 보여줄 때 사용자의 감성에 해당하는 라이브러리에서 랜덤으로 현대시 데이터를 추출해 감성값과 함께 호출한다. 사용자는 웹페이지에 호출된 감성값과 현대시를 만족, 보통, 불만족으로 평가할 수 있고, 불만족일 경우 해당 현대시가 5가지의 감성 중 새롭게 분류되어야 할 감성을 택하게 함으로써 해당 현대시의 라이브러리를 수정할 수 있다.

임계치 기준	감성
0<=predict score<0.2	매우 나쁨
0.2<=predict score<0.4	나쁨
0.4<=predict score<0.6	보통
0.6<=predict score<0.8	좋음
0.8<=predict score<=1	매우 좋음

표 1. 감성 분석 기준

3.4 결과

레이어가 과적합(overfitting)되지 않는 선에서 epochs 수를 6까지 늘려 모델의 성능을 평가한 결과, 89%의 정확도(acurracy)를 보였다.

현대시의 경우 그림 2처럼 매우 나쁨에 해당하는 데이터가 859개, 나쁨에 해당하는 데이터는 354개, 보통은 29개, 좋음은 352개, 매우 좋음은 1,406개로 분류되었다. 이를 좋음, 보통, 나쁨 크게 3가지로 다시 나눌 경우, 1,213개, 29개, 1,758개가 되는데 보통에 해당하는 값이 타 데이터에 비해 현저히 낮게 분류된 것을 알 수 있었다. 이는 좋음, 나쁨으로만 분류된 극성 NSMC 데이터를 바탕으로 학습시킨 모델을 사용했기 때문으로 보인다.

인공지능 커뮤니티 사이트인 페이스북 <한국 인공지능 커뮤니티> 등을 통해 일반 사용자들의 평가를 수집한 결과 표 2처럼 분석된 감성에 대한 만족도는 88%, 현대시에 대한 만족도는 72%로 나타났다 (n = 200).

감성에 비해 낮은 만족도를 보인 현대시는 불만족은 12%로 상대적으로 높은 불만족도를 보였다. 한용운의 <이별은 미의 창조>의 경우 '이별'과 '미'를 함께 나열함으로써 역설적인 표현으로 이별을 찬미하는 내용을 담고 있다. 따라서 5개의 라이브러리 중 '좋음' 또는 '매우 좋음' 쪽에 해당하는 것이 맞지만 '나쁨'에 분류되어 있었다. 이처럼 의도와는 다른 값을 반환하였지만, 사용자는 이를 긍정적으로 평가하여 라이브러리가 수정되지 않았다. 시의 비유적인 표현이나 상징적 표현 또는 역설적인 표현을 감성에 따라 바르게 분류하기 위해서는 별도의 과정이 필요한 것으로 보이나, 동시에 다양한 표현 때문에 여러 가지로 해석될 수 있으므로 약간의 오류는 독자가 자신의 상황에 맞춰 해석하고 이해할 수 있는 것으로 보인다.

수정된 라이브러리 중, 감동에 의한 '슬픔'이 '좋음' 또는 '매우 좋음'으로 분류되어 있어 불만족한 경우도 있었다. 이처럼 원인과 결과에 따라 같은 감성이라도 다른 의미를 지닐 때 분류의 오류가 다수 발생했으며, 독자 역시 불만족하는 상황이 발생하기도 했다.

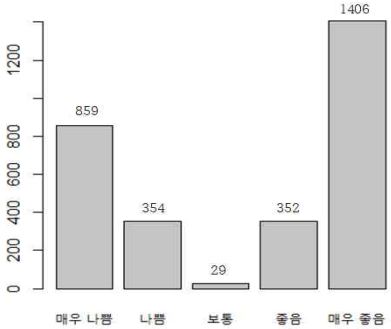

그림 2. 입력한 데이터에 대한 반환값

표 2. 각 데이터에 대한 만족도 평가

(단위: %)

평가 \ 데이터	감성	현대시
만족	88	72
보통	9	15
불만족	3	12

4. 결론 및 향후 과제

본 논문에서는 네이버 영화 리뷰 감성 데이터셋(NSMC)을 기반으로 학습한 koBERT 모델을 이용하여 사용자 입력에 따른 감성 분석을 시행하고, 사용자 입력에 적합한 현대시를 출력하는 웹 서비스 시스템을 구현하고 간단한 사용자 평가를 시행하였다.

감성 분석과 현대시에 대한 사용자 만족도 평가(샘플 크기 = 200)결과, 감성에 대한 만족도(88%)가 현대시에 대한 만족도(72%)보다 높게 나타났으며, 불만족도 또한 감성 분석(3%)이 현대시(12%)보다 매우 낮게 조사되었다. 보다 정확하고 높은 만족도를 위해 시의 비유적 표현, 역설적 표현과 같이 다양한 표현을 일반적으로 어떻게 해석해야 하는지 별도의 연구가 필요하다. 예를 들면, 원인과 결과에 따라 같은 감성이라도 다른 감성으로 해석되었던 감동에 의한 '슬픔'과 상실에 의한 '슬픔'처럼 이를 다르게 해석할 수 있도록 문맥을 고려한 분석 기준이 적용될 경우 보다 높은 정확도로 시 분류가 가능할 수 있다. 향후 연구로는 문맥 정보와 역설적 의미를 고려하여 감성 분석을 시행하고 이에 따른 시 분류를 적용할 계획이다.

참고 문헌

[1] 서상현, 김준태, "딥러닝 기반 감성분석 연구동향", 한국멀티미디어학회지, 20권, 3호, 8-22, 2016

[2] 오영택 외 2명, "Parallel Stacked Bidirectional LSTM 모델을 이용한 한국어 영화리뷰 감성 분석", 한국정보과학회, 46권, 1호, 45-49, 2019

[3] 이정훈, "감성분석 연구동향", 한국정보처리학회, 358-361, 2018

[4] Andrew Ng, "Machine Learning Yearning", 7장, 2018

[5] Naver Sentiment Movie Corpus (NSMC) https://github.com/e9t/NSMC

[6] Devlin, J., et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, In Proc. NAACL-HLT, 2019

[7] koBERT. https://github.com/SKTBrain/KoBERT

[8] 이영준, 최호진, "한국어 감정 분석을 위한 합동 학습 기반 KoBERT 모델", 한국정보과학회, 568-570, 2020

[9] Liyuan Liu, et al., "On The Variance Of the Adaptive Learning Rate And Beyond", In Proc. ICLR, 2020