감성 단어 등장 순서를 고려한 영화 리뷰 감성 분석

김홍진[O,1], 김담린[1], 김보은[2], 오신혁[1], 김학수[2]

강원대학교 컴퓨터정보통신공학전공[1], 건국대학교 인공지능학과[2]
jin3430@kangwon.ac.kr, ekafls33@naver.com, boeun@konkuk.ac.kr,
osh7605@kangwon.ac.kr, nlpdrkim@konkuk.ac.kr

Movie Revies Sentiment Analysis Considering the Order in which Sentiment Words Appear

Hong-Jin Kim[O,1], Dam-Rin Kim[1], Bo-Eun Kim[2], Shin-Hyeok Oh[1], Hark-Soo Kim[2]
Kangwon National University Department of Computer and Communications Engineering[1],
Konkuk University Department of Artificial Inteligence[2]

요 약

감성 분석은 문장의 감성을 분석해 긍정 또는 부정으로 분류하는 작업을 의미한다. 문장에 담긴 감성을 파악해야 하기 때문에 문장 전체를 이해하는 것이 중요하다. 그러나 한 문장에 긍정과 부정의 이중 극성이 동존하는 문장은 감성 분석에 혼동이 생길 수 있다. 본 논문에서는 이와 같은 문제를 해결하기 위해 단어의 감성 점수 예측을 통해 감성 단어 등장 순서를 고려한 감성 분석 모델을 제안한다. 또한 최근 다양한 자연어 처리 분야에서 좋은 성능을 보이는 사전 학습 언어 모델을 활용한다. 실험 결과 감성 분석 정확도 90.81%로 기존 모델들에 비해 가장 좋은 성능을 보였다.

주제어: 감성 분석, 언어 모델, 영화 리뷰

1. 서론

감성 분석(Sentiment Analysis)은 텍스트 데이터(Text Data)에서 감성을 분석해 긍정 또는 부정으로 분류하는 작업을 의미한다. 문장에 담긴 감성을 분류해야 하기 때문에 문장 전체를 이해하는 것이 중요하다[1-2]. 그러나 NSMC(Naver Sentiment Movie Corpus)[3]와 같은 한국어 감성 분석 데이터에는 실제 사람들이 입력한 리뷰이기 때문에 띄어쓰기, 맞춤법, 오타와 같은 문법적인 오류가 많아 문장을 정확하게 이해하기 어려운 문제가 있다. 또한 긍정과 부정의 이중 극성이 동존하는 문장은 감성 분석에 혼동이 생기는 문제가 있다. 아래의 표 1은 NSMC 데이터에서 긍정 단어와 부정단어가 동존하는 문장의 예시를 보여준다.

표 1 이중 극성이 동존하는 문장 예시

예시	감성
보면서 약간 **실망했지만** 마지막이 넘 **좋았다**	긍정
내용 진짜 **뻔한데 볼만함**	긍정
이 **막장스토리에** 배우들 연기는 정말 **잘하네**	긍정
처음엔 **괜찮았는데** 갈수록 보기 **싫어**	부정
좋은 영화다. 하지만 **쓰레기다.**	부정

위의 표 1에서 굵게 표시된 부분은 긍정 또는 부정을 나타내는 어절이다. 예시 문장의 감성이 처음에 나온 감성에서 마지막에 나온 감성으로 반전되는 것을 확인할 수 있다. 본 논문에서는 문법적 오류를 보완하기 위해 띄어쓰기 모듈[4]을 이용해 정제를 거치고, 표 1과 같이 이중 극성이 동존하는 문장에서 감성 분석 혼동의 문제를 보완하기 위해 감성 단어 등장 순서를 고려한 감성 분석 방법을 제안한다.

2. 관련 연구

최근 감성 분석 연구는 딥러닝을 활용하는 방법이 주로 수행되었다. RNN(Recurrent Neural Network)과 CNN(Convolutional Neural Network)에 기반한 연구가 수행되었고 좋은 성능을 보였다[5-8]. [5]는 발화자의 의도를 파악하기 위해 CNN을 이용한 화행, 술어, 감성을 동시에 식별하는 통합 모델을 제안했다. [6]은 오타를 포함한 구어체에 강건한 형태소, 음절, 자소를 동시에 고려하는 Multi-channel CNN 모델을 제안했다. [7]은 LSTM(Long Short-Term Memory)의 셀(Cell) 크기를 다르게 주어 병렬적으로 연결한 Parallel stacked Bidirectional LSTM 모델 구조를 제안했다. [8]은 LSTM 기반의 언어 모델(Language Model) 중 하나인 ELMo(Embedding from Language Model)을 활용하는 감성 분석 모델을 제안했다. [1]은 NSMC에 포함된 문법적 오류를 최소화하기 위한 전처리 방법으로, 띄어쓰기 정제를 거쳐 형태소 분석의 결과로 나오는 분석불능범주인 형태소를 음절로 치환하는 방법을 제안했다. 한편, 언어 모델을 대용량 코퍼스로 학습하여 이용한 연구들이 다양한 자연어처리 분야에서 높은 성능 향상을 보이고 있다[2, 9]. [2]는 양방향성을 가진 트랜스포머(Transformer)[10]을 기반으로, 셀프 어텐션 매커니즘(Self-Attention Mechanism)과 문장에서 임

의의 단어를 마스킹(Masing)하고 예측하도록 학습한 BERT(Bidirectional Encoder Representation from Transformers)[11]를 활용하여 감성 분석을 수행했다. [9]는 BERT 모델에서 매 학습 때마다 임의의 단어가 동적으로 마스킹이 될 수 있도록 개선된 RoBERTa[12]를 활용하여 감성 분석을 수행했다. BERT의 후속 연구로 ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately)[13]는 Generator에서 임의의 단어를 마스킹하고 예측하도록 학습한 다음, Discriminator에서 생성한 단어 열에 대해서 각 단어가 원래 입력과 동일한 것인지 치환된 것인지 예측하도록 학습한다. 본 논문에서는 기존 연구에 착안하여 띄어쓰기 모듈을 통해 NSMC 데이터를 정제하고, ELECTRA를 활용하여 감성 단어 등장 순서를 고려한 감성 분석 모델을 제안한다.

3. 감성 단어 등장 순서를 고려한 감성 분석 모델

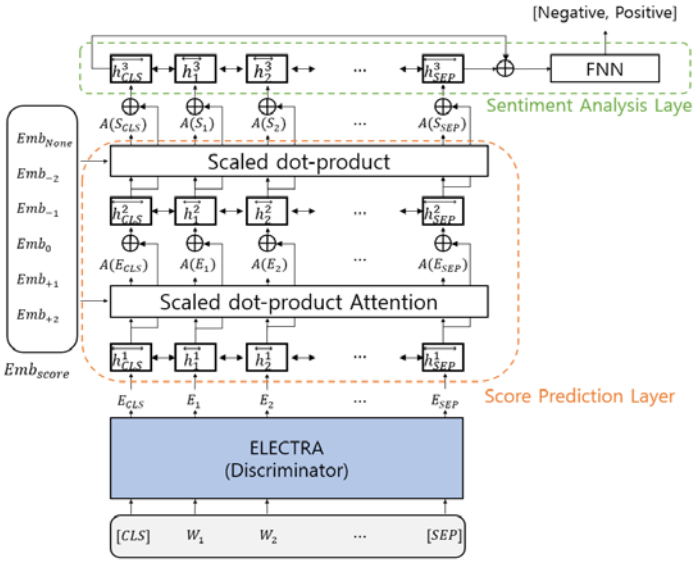

그림 1 감성 분석 모델 전체 구조도

그림 1은 본 논문에서 제안하는 감성 분석 모델 전체 구조이며 크게 ELECTRA, 감성 점수 예측 계층, 감성 분석 계층으로 구성되어 있다. ELECTRA에서 입력 데이터를 받아 단어 표현 벡터(Vector)를 생성하고 감성 점수 예측 계층에서 각 단어에 대한 감성 점수를 학습하며, 마지막으로 감성 분석 계층에서는 감성 점수 예측 계층에서 출력된 벡터를 입력으로 사용해 감성 단어 등장 순서를 반영하여 감성 분석을 수행한다.

3.1 감성 점수 부착

문장에서 어떤 단어가 긍정 또는 부정 단어인지 예측하도록 학습하기 위해서는 긍정, 부정 단어가 태깅(Tagging)되어 있어야한다. 본 논문에서는 NSMC 데이터에 감성 점수를 부착하기 위해서 [14]의 감성 사전을 활용했다. 아래 표 2는 감성 점수의 기준을 보여준다.

표 2 감성 점수 기준

감성 점수	감성
해당 없음	해당 없음
-2	매우 부정
-1	부정
0	중립
+1	긍정
+2	매우 긍정

아래 표 3은 표 2의 기준으로 감성 점수를 부착한 문장의 예시를 보여준다.

표 3 감성 점수 부착 예시

예시
보면서 약간 **실망했지만{-2}** 마지막이 넘 **좋았다{+2}**
내용 진짜 **뻔한데{-1}** **볼만함{+1}**
이 **막장스토리에{-2}** 배우들 연기는 정말 **잘하네{+2}**
처음엔 **괜찮았는데{+1}** 갈수록 보기 **싫어{-2}**
좋은{+2} 영화다. 하지만 **쓰레기다{-2}.**

표 3에서 굵게 표시된 부분이 긍정 또는 부정 단어에 해당되어 감성 점수가 부착된 것을 볼 수 있다.

3.2 감성 점수 예측 계층

감성 점수 예측 계층에서는 LAN(Label Attention Network)[15]의 구조를 2개의 계층으로 구성하여 단어 벡터와 감성 점수 임베딩(Embedding) 벡터의 연관성을 계산하여 각 단어에 대한 감성 점수를 예측한다. 그림 1에서 E_i는 ELECTRA의 출력인 단어 표현 벡터이다. 첫번째 계층에서는 문맥 정보를 반영하기 위해 양방향 LSTM으로 인코딩한 후 연결하여 사용하며 수식은 다음과 같다.

$$\begin{aligned} \overrightarrow{h_i^1} &= LSTM(E_i, \overrightarrow{h_{i-1}^1}) \\ \overleftarrow{h_i^1} &= LSTM(E_i, \overleftarrow{h_{i-1}^1}) \\ \vec{h}_i^1 &= [\overrightarrow{h_i^1}; \overleftarrow{h_i^1}] \\ \vec{H}^1 &= \{\vec{h}_{CLS}^1, \vec{h}_1^1, \dots, \vec{h}_{SEP}^1\} \end{aligned} \quad (1)$$

수식 (1)에서 $\overrightarrow{h_i^1}$는 정방향 은닉 상태(Forward Hidden State)이며 $\overleftarrow{h_i^1}$는 역방향(Backward) 은닉 상태이다. $\vec{h}_i^1$는 i번째 단계에서 양방향 은닉 상태를 연결한 벡터이다. $\vec{H}^1$는 양방향 문맥 정보가 반영된 각 단어를 나타내는 벡터이다. 다음으로, $\vec{H}^1$과 그림 1의 감성 점수 임베딩인 $Emb_{score} = \{Emb_{None}, Emb_{-2}, \dots, Emb_{+2}\}$ 사이의 연관성을 계산하기 위해 Scaled dot-Product Attention을 사용하며 수식은 다음과 같다.

$$\begin{aligned} A(E_i) &= Attention(Q, K, V) = \alpha * V \\ Q &= \vec{H}^1, K = V = Emb_{score} \end{aligned} \quad (2)$$

$$\alpha = softmax(\frac{Q * K^T}{\sqrt{d_h}})$$

수식 (2)에서 사용한 감성 점수 임베딩은 랜덤 초기화(Random Initialize)하여 사용했으며, 학습 과정에서 조정(Tuning)된다. d_h는 정규화 값이며 감성 점수 임베딩 크기와 동일하다. 또한 $A(E_i)$는 어텐션 벡터이며 각 단어에 대한 감성 점수 분포가 강조된 벡터를 나타낸다. 두 번째 계층에서는 첫번째 LSTM의 출력인 $\vec{H}^1$와 어텐션 벡터 $A(E_i)$를 연결(Concatenation)하여 수식 (1)과 같은 과정으로 인코딩하며 수식은 다음과 같다.

$$\begin{gathered}\vec{h}_i^2 = LSTM([\vec{H}^1; A(E_i)], \vec{h}_{i-1}^2) \\ \overleftarrow{h}_i^2 = LSTM([\vec{H}^1; A(E_i)], \overleftarrow{h}_{i-1}^2) \\ \overleftrightarrow{h}_i^2 = [\vec{h}_i^2; \overleftarrow{h}_i^2] \\ \overleftrightarrow{H}^2 = \{\overleftrightarrow{h}_{CLS}^2, \overleftrightarrow{h}_1^2, \dots, \overleftrightarrow{h}_{SEP}^2\}\end{gathered} \quad (3)$$

다음으로, $\overleftrightarrow{H}^2$ 과 Emb_{score} 에 대하여 Scaled dot-Product를 수행하며 수식은 다음과 같다.

$$\begin{gathered}A(S_i) = \frac{Q * K^T}{\sqrt{d_h}} \\ Q = \overleftrightarrow{H}^2, K = Emb_{score}\end{gathered} \quad (4)$$

수식 (4)의 $A(S_i)$를 이용하여 각 단어에 대한 감성 점수는 다음과 같이 예측된다.

$$\widehat{Score} = argmax(softmax(A(S_i))) \quad (5)$$

3.3 감성 분석 계층

감성 분석 계층에서는 감성 단어 등장 순서를 반영하기 위해 감성 점수 예측 계층의 마지막 출력인 $\overleftrightarrow{H}^2$와 $A(S_i)$를 연결하여 양방향 LSTM으로 인코딩하며 수식은 다음과 같다.

$$\begin{gathered}\vec{h}_i^3 = LSTM([\overleftrightarrow{H}^2; A(S_i)], \vec{h}_{i-1}^3) \\ \overleftarrow{h}_i^3 = LSTM([\overleftrightarrow{H}^2; A(S_i)], \overleftarrow{h}_{i-1}^3) \\ \overleftrightarrow{H}^3 = [\vec{h}_{SEP}^3; \overleftarrow{h}_{CLS}^3]\end{gathered} \quad (6)$$

수식 (6)의 $\overleftrightarrow{H}^3$은 정방향의 마지막 은닉 상태와 역방향의 마지막 은닉 상태를 연결한 값이며 감성 단어 등장 순서가 반영된 문맥 벡터이다. 최종적으로 긍정, 부정을 분류하기 위해 $\overleftrightarrow{H}^3$는 FNN(Feed-forward Neural Network)을 거쳐 감성 분석 결과를 출력한다.

$$\hat{Y} = FNN(\overleftrightarrow{H}^3) \quad (7)$$

3.4 학습 방법

본 논문의 감성 분석 모델은 감성 점수 예측과 감성 분석을 동시에 학습한다. 감성 점수 예측 계층에서는 각 단어에 대하여 예측한 감성 점수와 정답 감성 점수 간의 크로스 엔트로피(Cross-entropy)를 최소화하도록 학습하며 수식은 다음과 같다.

$$H_{\widehat{score}} = -\sum_i \widehat{Score}_i \log(Score_i) \quad (8)$$

감성 분석 계층에서는 모델이 예측한 감성과 정답 감성 간의 크로스 엔트로피를 최소화하도록 학습하며 수식은 다음과 같다.

$$H_{\hat{Y}} = -\sum_n \hat{Y}_n \log(Y_n) \quad (9)$$

본 논문에서는 매 학습 마다 감성 분석 비용 함수는 0.9, 감성 단어 예측 비용 함수는 0.1만큼 비율이 반영되도록 설정했다.

4. 실험 및 결과

본 논문에서 사용한 ELECTRA는 20GB의 한국어 위키피디아, 뉴스 데이터를 사용하여 사전 학습한 모델이다. 또한 실험 데이터로 NSMC 데이터를 사용하였으며 학습 데이터는 15만개, 평가 데이터는 5만개로 구성되어 있다. NSMC 데이터에 포함된 띄어쓰기 오류를 최소화하기 위해 [4]를 이용하여 띄어쓰기 교정 후 사용했다. 표 4은 본 논문에서 사전 학습한 ELECTRA의 활용에 따른 NSMC 데이터의 성능 비교이다.

표 4 ELECTRA 활용에 따른 NSMC 성능 비교

Model	Accuracy
ELECTRA + FNN (Ours)	90.38
ELECTRA + LSTM (Ours)	90.57
ELECTRA + SP + SA (Ours)	**90.81**

표 1에서 ELECTRA+FNN은 ELECTRA 출력 중 E_{CLS} 벡터에 FNN을 적용하여 감성 분석을 수행한 모델이며, ELECTRA + LSTM은 ELECTRA 출력을 양방향 LSTM으로 인코딩 후 감성 분석을 수행한 모델이다. ELECTRA+SP+SA는 본 논문에서 제안하는 방법인 ELECTRA와 감성 단어 예측 계층 그리고 감성 단어 등장 순서를 고려한 감성 분석 계층을 모두 적용한 모델이다. 실험 결과, 본 논문에서 제안한 감성 단어 등장 순서를 고려한 감성 분석 모델이 가장 좋은 성능을 보였다.

표 5는 본 논문에서 제안하는 모델과 기존 감성 분석 모델들에 대한 NSMC 데이터 성능 비교이다.

표 5 기존 모델들과 NSMC 데이터 성능 비교

Model	Accuracy
Multi-channel CNN [6]	86.27
Parallel stacked Bi-LSTM [7]	88.95
RNN + ELMo [8]	89.24
RoBERTa [9]	89.88
BERT + RNN [2]	90.51
ELECTRA + SP + SA (Ours)	**90.81**

표 5의 RoBERTa[9]는 RoBERTa의 출력을 양방향 LSTM으로 인코딩 한 후 감성 분석을 수행한 모델이다. 또한 BERT+RNN[2]는 RNN 출력을 양방향 RNN으로 인코딩 한 후 감성 분석을 수행한 모델이다. 실험 결과, 본 논문에서 제안한 감성 단어 등장 순서를 고려한 감성 분석 모델이 가장 좋은 성능을 보였다. 이는 감성 점수 예측 계층의 출력 값을 통해 감성 단어 등장 순서를 고려하여 문맥 정보를 반영하는 방법이 효과적임을 보인다.

5. 결론 및 향후연구

본 논문에서는 언어 모델인 ELECTRA를 한국어 대용량 말뭉치로 사전 학습하여 활용하고, 단어의 감성 점수 예측 계층을 설계하여 감성 단어 등장 순서를 반영하는 감성 분석 모델을 제안한다. 한국어 감성 분석 데이터 NSMC 실험 결과, 본 논문에서 제안하는 모델이 기존 모델보다 좋은 성능을 보였다. 향후 연구로, 본 모델에서 사용한 감성 점수 임베딩을 기쁨, 슬픔, 분노와 같은 더 구체적인 감정 임베딩으로 대체하여 문장에 나타난 감정을 반영하는 감성 분석 모델을 실험할 예정이다.

감사의 글

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2020-0-00368, 뉴럴-심볼릭(neural-symbolic) 모델의 지식 학습 및 추론 기술 개발)

참고문헌

[1] 김담린, 김보은, 장영진, 권오욱, 김학수, "맞춤법 오류를 포함한 영화 리뷰 감성 분석의 노이즈를 줄이기 위한 처리 방법", 한국정보과학회 2020년 한국컴퓨터종합학술대회 논문집, pp. 1447-1449, 2020.

[2] 박천음, 이창기, "BERT 기반 Variational Inference와 RNN을 이용한 한국어 영화평 감성 분석", 정보과학회 컴퓨팅의 실제 논문지, 제25권, 제11호, pp. 552-558, 2019.

[3] L. Park. Naver Sentiment Movie Corpus v1 [Online]. Available: https://github.com/e9t/nsmc (downloaded 2018, Aug. 3)

[4] H. Kim and H. Kim, "Effective Integration of Automatic Word Spacing and Morphological Analysis in Korean", 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 275-278, 2020.

[5] M. Kim and H. Kim, "Integrated neural network model for identifying speech acts, predicators, and sentiments of dialogue utterances", Pattern Recognition Letters, Vol. 101, pp. 1-5, 2018.

[6] 김민, 변증현, 이충희, 이연수, "Multi-channel CNN을 이용한 한국어 감성분석", 제30회 한글 및 한국어 정보처리 학술대회 논문집, pp. 79-83, 2018.

[7] 오영택, 김민택, 김우주, "Parallel Stacked Bidirectional LSTM 모델을 이용한 한국어 영화리뷰 감성 분석", 정보과학회논문지, 제46권, 제1호, pp. 45-49, 2019.

[8] 박천음, 김건영, 황현선, 이창기, "문맥 표현 기반 한국어 영화평 감성 분석", 제30회 한글 및 한국어 정보처리 학술대회 논문집, pp. 75-78, 2018.

[9] 민진우, 나승훈, 신종훈, 김영길, "RoBERTa를 이용한 한국어 자연어처리: 개체명 인식, 감성분석, 의존파싱", 한국정보과학회 한국소프트웨어종합학술대회 논문집, pp. 407-409. 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all You Need", Neural Information Processing Systems (NIPS), pp. 5998-6008, 2017.

[11] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (1), 2019.

[12] Y. Liu, M. Ott, Na. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", arxiv.org/abs/1907.11692, 2019.

[13] K. Clark, M. T. Luong, Q. V. Le and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators", International Conference on Learning Representations (ICLR), 2019.

[14] 온병원, 박상민, 나철원, KnuSentiLex [Online]. Available: https://github.com/park1200656/KnuSentiLex, 2018.

[15] C. Leyang and Y. Zhang, "Hierarchically-Refined Label Attention Network for Sequence Labeling", Empirical Methods in Natural Language Processing (EMNLP) and International Joint Conference on Natural Language Processing (IJCNLP), pp. 4106-4119, 2019.