|  |
| --- |
| **LSTM을 이용한 감정 분석**  김건희O, 한윤, 임특  고려대학교  {hobbyholic, hyku2014}@korea.ac.kr, linte960326@gmail.com  **Sentiment Classification using LSTM**  Geonhui KimO, Yoon Han, ~  Korea University  **요 약**  본 논문에서는 LSTM(Long Short-Term Memory)을 이용한 감성 분석 방법을 제안한다. 실험 데이터는 한국어의 경우 네이버 영화 사이트에서의 영화평을 추출한 NSMC(Naver Sentiment Movie Corpus)로 총 15,000건이며, 영어의 경우 드라마 프렌즈(Friends)의 대본으로 총 10,561개의 문장으로 이루어져 있다. 실험에 사용한 딥러닝 모델은 한국어 데이터의 경우에는 Bi-LSTM, 영어 데이터의 경우에는 LSTM을 이용하여 학습을 수행하였으며 평과 결과, 한국어 감정 분석 모델은 약 85%, 영어 감정 분석 모델은 약 41%의 정확도를 보였다.  주제어 : 감정 분석, LSTM, Bi-LSTM |

**1. 서론**

스마트기기와 소셜 미디어(SNS)의 활성화로 많은 사람들의 의견을 표현하는 글들이 시공간의 제약에서 벗어나 손쉽게 작성되고 기하급수적으로 쌓이고 있다. 그에 따라 의견이나 감정을 파악하는 **오피니언 마이닝**(Opinion Mining)에 관한 연구들이 이루어지고, 기업의 경우 사용자의 주관적인 의견과 견해를 담고있는 댓글이나 이모티콘 등의 데이터를 적극적으로 수집, 분석하여 자사의 제품이나 서비스에서 개선해야 할 부분에 대한 인사이트(Insight)를 얻는 등 ‘감성 분석’의 중요성이 커지고 있다. 이처럼 오피니언 마이닝은 대중의 의견, 인사이트를 필요로 하는 기업 마케팅이나 정치 등의 분야에서 사용할 수 있는 기술이기 때문에 많은 가능성을 가지고 있다.

앞서 언급한 오피니언 마이닝은 어떤 사람이나 인물에관한 사람들의 주관적인 의견을 수치화하여 객관적인 정보로 바꾸는 기술이다. 분석 대상이 텍스트이기 때문에 텍스트 마이닝(Text Mining)의 한 분야이며 텍스트 마이닝에서 활용하는 자연어 처리(NLP)방법을 사용한다. 또한 텍스트에서 의견뿐 만 아니라 감정과 태도도 분석하기 때문에 감성 분석(Sentiment Analysis) 또는 감성 분류(Sentiment Classification)로도 불린다. 오피니언 마이닝은 3개의 단계를 거치는데 사실과 의견을 구분하여 어휘 정보를 추출한 후, 그 어휘가 해당 텍스트에서 어떤 의미로 사용되었는지에 대한 판단 및 분류 단계를 거친다. 이때 크게 긍정이나 부정과 같은 감정으로 판단 및 분류하며, 이와 같이 분석된 정보들은 요약되어 사용자에게 전달되어진다.

감정을 분류하는 방법에 있어서는 감정 사전 구축을 통하여 긍정/부정 가중치를 계산하는 **규칙 기반 방법**과 긍정/부정이 포함되어 있는지를 판별하는 **확률 기반 방법**으로 크게 나뉜다. 감정 사전을 활용하는 경우 감정과 관련된 단어나 형태소의 빈도수를 구하여 판단한다. 하지만 문법적으로 맞지 않거나, 사전에 없는 단어가 출현하였을 때 분석하기 힘들다는 단점이 있어 기계학습을 이용한 확률 기반 방법이 많이 이용되고 있다.

본 논문에서는 영어 데이터셋과 달리 한국어 데이터셋의 경우 MeCab을 이용하여 형태소 분석을 한 후 토큰화를 진행하는 방법으로 문장을 처리하였다. 이후 한국어 데이터셋은 Bi-LSTM 모델을, 영어 데이터셋은 LSTM 모델을 적용하여 학습을 하였다. 실험 결과 약 85%, 영어 감정 분석 모델은 약 41%의 정확도를.

본 논문의 구성은 다음과 같다. 2장에서는 학습 모델로 활용한 LSTM에 대해서 기술한다. 3장에서는 2장에서 제시한 내용을 이용하여 실험한 내용을 기술한다. 마지막으로 4장에서는 결론에 대해 논의하고 논문을 마친다.

**2. 모델 설명**

이 장에서는 RNN(Recurrent Neural Network)에 대하여 간단히 설명하고, 본 논문에서 이용한 LSTM을 이용한 학습 방법 그리고 더 나아가 LSTM을 양방향으로 적용한 bi-LSTM에 대하여 서술한다.

**2.1 LSTM(Long Short-Term Memory)**

RNN은 반복적인 데이터, 순차적인 데이터(Sequential Data)를 학습하는데 특화되어 발전한 인공신경망의 한 종류이다. 순차적인 데이터의 가장 대표적인 것이 언어로, 단어 뒤에 단어가 연속적으로 제공되는 데이터이기 때문에 음성인식, 단어 의미 파악 등의 자연어 처리(Natural Language Process)에 이용된다. 기존의 인공신경망(Neural Network)은 입력된 데이터만 이용하여 학습한다는 점과 비교하였을 때, RNN에서 두드러지는 특징은 이전에 학습된 정보를 기억하고 다음 학습에 반영한다는 점이다. RNN이 학습하는 과정은 (그림 1)과 같다.

(그림 1) RNN 모델[[1]](#footnote-1)

|  |
| --- |
| Figure 1 Recurrent Neural Network |

이론상 RNN은 순차적 데이터의 첫번째 항부터 마지막 항까지 학습이 가능한 것처럼 보인다. 하지만 실제 학습을 실행해보면 Time Step가 길수록 이전의 정보를 기억하지 못하는 것을 발견할 수 있는데, 이를 **장기 의존성**(Long-Term Dependency)문제라고 한다. 이는 모델의 학습과정에서 출력과 멀리 떨어진 Time Step일 수록 역전파(Back Propagation)가 전달되지 않는 Vanishing Gradient로부터 기인한다.

이를 해결하기 위한 방법으로 **LSTM**이 Hochreiter& Schmidhuber에 의해 소개되었다. LSTM 모델 학습의 과정은 (그림 2)와 같다.

(그림 2) LSTM 모델[[2]](#footnote-2)

|  |
| --- |
| Figure 2 Long Short-Term Memory Model |

LSTM에서 가장 핵심적인 부분은 **Cell State**로, 위 그림에서 데이터의 흐름 중 가장 위에 있는 선에 해당한다. Cell State는 곱셈과 덧셈으로만, Linear Interaction만을 적용시키기 때문에 이전 단계에서 전달되는 정보가 전혀 바뀌지 않고 그대로 흐르게 하는 것을 가능하게 한다.

LSTM의 첫 단계로는 어떤 정보를 버릴 것인지 정하는 단계로 Forget Gate Layer라고 부른다. 이 단계에서는 이전 Time Step에서 전달된 값과 입력된 를 받아 Sigmoid Layer에 의해 값을 결정한다. 여기서 0이 반환되는 경우 곱셈에 의해서 기존의 모든 정보이 제거된다. 반대로 1이 반환되는 경우 이 그대로 보존되게 된다.

두번째 단계에서는 이후 들어오는 새로운 정보 중 어떤 것을 Cell State에 저장할 것인지를 결정한다. 먼저, Input Gate Layer라고 불리는 Sigmoid Layer에서 어떤 값을 업데이트를 할 것인지 정한다. 이후 tanh Layer가 새로운 후보 값들인 를 만들어 두 값을 곱한 값을 더해 새로운 Cell State인 를 생성하게 된다.

마지막 단계에서는 무엇을 출력할지 정하는 단계인데, 이 값은 Cell State를 적용하여 필터링 된 값으로 결정된다.

**2.2 Bi-LSTM**

RNN이나 LSTM은 입력 순서를 시간 순서대로 입력하기 때문에 결과물이 직전 패턴을 기반으로 수렴하는 경향을 보인다는 한계점을 가지고 있다. 이 단점을 해결하기 위해 역방향을 추가하여 입력 문장의 단어를 반대로도 처리하게 한 모델이 바로 Bidirectional LSTM이다. 또한 단어와 구(Phrase) 간 유사성을 입력벡터에 내재화해 성능을 개선하였으며 그 구조는 (그림 3)과 같다.

(그림 3) Bi-LSTM 모델[[3]](#footnote-3)

|  |
| --- |
| Figure Bidirectional-LSTM Model |

**3. 실험 및 결과**

**3.1 한국어 데이터 전처리**

한국어 데이터셋으로 사용된 NSMC는 네이버 영화 서비스에서 생성된 영화평을 수집하여 생성되었다. 영화 리뷰에 포함된 감정을 보다 확실하게 구분하기 위해 수집된 데이터 중 평점이 1~4점인 영화평은 부정, 9~10점인 영화평은 긍정으로 하였다. 긍정/부정 리뷰의 비율은 동일하게 샘플링 되었으며 그 결과는 <표 1>과 같다.

<표 1> 긍정/부정 영화평 빈도수

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Training Data | 75170 | 74825 |
| Test Data | 24826 | 25171 |

샘플링 된 각 리뷰 데이터들은 MeCab을 이용하여 품사 태깅(POS tagging, Part-Of-Speech Tagging)을 진행하였다. 이후, 각 영화 리뷰에 토큰 리스트에 들어있는 토큰을 숫자로 변환하는 Text-To-Sequence 과정을 진행하였다. 실험에 사용할 LSTM 모델의 경우 입력 데이터의 길이가 Timestep과 동일해야 하기 때문에 토큰의 개수가 적은 경우에는 맨 앞에 0을 추가하였다. 반대로, 토큰의 개수가 더 많은 경우에는 뒷부분을 제거하는 방법으로 전처리를 마무리하였으며 그 예시는 (그림 3)과 같다.

(그림 4) Padding 설정 예시

|  |
| --- |
|  |

**3.2 영어 데이터 전처리**

영어 데이터셋으로 사용된 드라마 프렌즈(Friends)의 자막을 수집하여 생성되었다. 영어 데이터셋의 가장 두드러지는 특징은 한국어 데이터셋과 다르게 긍정/부정이 아닌 2개 이상의 다양한 감정이 Labeling 되어 제공되었다는 데 있다.

자막에 나타난 드라마 대사의 경우 품사 태깅과 같은 과정은 거치지 않고 Tokenize를 통해 문장을 Sequence로 전환하였다. 또한, 앞에서 한국어 데이터 전처리를 하는 방식과 같이 각 문장의 길이와 Timestep의 크기를 비교하여 Padding 또는 Cropping을 진행하여 대사에 대한 전처리는 마무리하였다.

마지막으로 2개 이상의 감정이 Labeling 되어있기 때문에 LSTM 모델을 이용하기 위해서는 출력값을 Metric Variable로 전환해야 한다. 따라서 0~8까지 Labeling된 출력 값을 One Hot Encoding 방식을 사용하여 적절한 형태로 변환해 주었고 그 결과는 (그림 5)와 같다.

(그림 5) One Hot Coding 예시

|  |
| --- |
|  |

**3.3 모델 생성 및 학습**

한국어 모델을 Bi-LSTM로 설정하기 전, 차이를 비교하기 위해 LSTM 모델로 학습을 진행하였다. 학습에 사용된 모델의 형태는 데이터셋에 따라 (그림 6)과 같이 생성하였다.

(그림6) 학습 모델

|  |
| --- |
|  |

한국어 모델의 학습 과정에서 손실 함수는 Binary Cross Entropy, 최적화 방법은 Adam, 훈련 횟수는 10번으로 설정하였다. 영어 모델은 손실 함수로 Categorical Cross Entropy로, 훈련 횟수는 50번으로 설정한 후 학습을 진행하였다.

**3.4 학습 결과**

한국어 모델은 약 50,000개의 데이터셋으로, 영어 모델은 3,296개의 데이터셋으로 테스트를 진행하였다. 그 결과는 <표 2>와 같다.

<표 2> 테스트 결과

|  |  |  |
| --- | --- | --- |
|  | 한국어 | 영어 |
| Test Loss | 0.7438 | 2.6622 |
| Test Accuracy | 0.8391 | 0.4365 |

**4. 결론**

한국어 모델을 LSTM으로 설정했을 때와 Bi-LSTM으로 설정했을 때 정확도 차이가 0.2%로 무의미한 차이가 나나는 것 실험 중 확인하였다. 이는 감정 분석에서 데이터 전처리의 영향이 더 크다는 점, 데이터셋의 토큰 리스트 길이가 길지 않아 모델을 바꾸어도 영향을 받지 않았다는 점, 영화 리뷰 특성 상 앞뒤로 연결되는 대명사와 같은 단어가 적다는 점에서 Bi-LSTM의 효과가 크게 나타나지 않은 것이라고 판단하였다.

더 높은 정확도를 높이기 위해서는 감정 사전 기반 점수 생성과 같은 데이터 전처리를 추가하거나 BERT와 같은 Pre-trained 모델을 이용해보는 연구가 더 필요하다.

**참고문헌**

1. “오피니언 마이닝의 이해.”, Naver blog. Last modified June 11, 2016, accessed June 28, http://blog.naver.com/zxy826/220733365086
2. “Understanding LSTM Networks. Colah’s blog. Last modified August 27, 2015, accessed June 28, http://colah.github.io/posts/2015-08-Understanding-LSTMs/
3. “Bi-LSTM.”Medium. last modified July 4, 2019, accessed June 28, http://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0

1. , 2 <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> [↑](#footnote-ref-1)
2. 3 <http://www.gabormelli.com/RKB/Bidirectional_LSTM_(biLSTM)_Model> [↑](#footnote-ref-2)
3. [↑](#footnote-ref-3)