



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

주목 기반 순환신경망을 이용한
감성 분석과 활용

2018년 2월

서울대학교 공학전문대학원

응용공학과

주현탁

Sentiment Analysis and Its Applications with Attention-based Recurrent Neural Network

지도 교수 성 원 용

이 논문을 공학석사 학위논문으로 제출함

2018년 2월

서울대학교 공학전문대학원

응용공학과

주현탁

주현탁의 공학석사 학위논문을 인준함

2018년 2월

위원장 _____ (인)

위원 _____ (인)

위원 _____ (인)

초 록

주목 기반 순환신경망을 이용한 감성 분석과 활용

감성 분석은 주어진 글이 가리키는 대상이나 화자의 의견 또는 평가를 분석하는 분야로 글의 주체가 가진 의견이나 감성은 수치적으로 파악하기 어렵기 때문에 자연어 처리에서도 어려운 분야로 여겨져 왔다. 감성 분석 분야는 최근 급부상한 빅데이터의 열풍과 함께 선거나 광고, 마케팅 등의 영역에서 사람의 직관에 의존하던 부분들을 대체하여 체계적이고 정밀한 분석을 하는 데에 큰 성과를 이루고 있다.

본 연구는 네이버 영화 리뷰 데이터와 Rotten Tomato Movie Review 데이터를 이용하여 순환신경망 모델을 중심으로 글이 긍정적인 판단을 내리고 있는지 부정적인 판단을 내리고 있는지를 효과적으로 분류하는 방법에 대한 내용을 다룬다. 텍스트를 컴퓨터가 이해할 수 있는 형태인 벡터로 변환하는 과정에서 문서를 자모, 글자, 단어, 형태소로 나누는 방법을 제안하고 각각을 감성 분석 모델의 입력으로 사용할 때 어떠한 변환 방법이 가장 좋은 성능을 내는지를 비교해 본다.

감성 분석 모델로는 기존의 순환신경망을 개선한 LSTM, Bi-directional LSTM 과 여기에 Attention Mechanism 를 적용한 모델을 사용하였고 이것을 비 신경망 모델인 Naïve Bayes Classifier 와 비교해 보았다.

네이버 영화 리뷰 데이터에서는 형태소 기반으로 문서를 나누었을 때 Bi-directional LSTM 에 Attention Mechanism 을 적용한 모델이 가장 우수한 성능을 보였고 Rotten Tomato Movie Review 데이터에서는 단어기반으로 Bi-directional LSTM 에 Attention Mechanism 을 적용한 모델이 가장 우수한 성능을 보이는 것을 확인하였다. 성능의 차이가 발생하는것은 훈련 데이터에 등록되지 않은 검증 데이터의 미등록어 비율이 주요한 원인으로 작용하는 것을 확인하였고 특히 한글의 경우 단어 단위로 문서를 나누었을 때 미등록어 비율이 현저하게 높아지는 것을 확인할 수 있었다.

분류와 더불어 Attention Mechanism 모델에서 각 문서에 대해 시퀀스의 어떤 시간 스텝에서 Attention Vector 가 활성화되는지를 분석하였고 이를 바탕으로 어떠한 형태소가 감정 판단을 하는데 영향을 끼치는지 정량적으로 분석할 수 있었다.

주요어: 감성분석, 순환신경망, 자연어처리

학번 : 2016-22258

목 차

제 1 장 서 론	1
제 2 장 관련 연구	4
2.1. 순환신경망	4
2.2. LSTM	7
2.3. Bi-directional LSTM	9
2.4. Attention Mechanism	11
제 3 장 한국어의 RNN 을 이용한 감성 분석	15
3.1. 한국어의 벡터 표현 방식	15
3.2. 형태소 분석	17
제 4 장 실험 및 결과	19
4.1. 실험 데이터	19
4.2. 실험 방법	24
4.3. 실험 결과	27
제 5 장 결론	34

참고 문헌 35

Abstract 40

표 목 차

〈표 1〉 ‘재밌다.’라는 문서를 다양한 문서의 구성 요소로 쪼갠 결과	15
〈표 2〉 자모 단위로 분할한 글자에 매핑된 사전	15
〈표 3〉 문서를 자모 단위로 나눈 뒤, 매핑 후 임베딩하는 과정	16
〈표 4〉 잘못된 형태소 분석의 예	18
〈표 5〉 네이버 리뷰 데이터에 포함된 비문과 예시	21
〈표 6〉 네이버 리뷰 에디터의 표현 방식에 따른 사전의 크기와 최대 시퀀스 크기	22
〈표 7〉 Rotten Tomato Movie Review 의 표현 방식에 따른 사전의 크기와 최대 시퀀스 크기	24
〈표 8〉 이진 분류에서 사용하는 혼동 행렬	26
〈표 9〉 네이버 영화 리뷰 데이터의 RNN 및 Naïve Bayes 모델의 정확도	28
〈표 10〉 Rotten Tomato Movie Review 의 RNN 및 Naïve Bayes 모델의 정확도	28
〈표 11〉 네이버 영화 리뷰 데이터의 미등록어 비율	30
〈표 12〉 Rotten Tomato Movie Review 의 미등록어 비율	30
〈표 13〉 입력 문서에 대한 Attention 벡터의 활성화에 대한 시각화	31
〈표 14〉 상위 10개의 Attention 을 갖는 형태소	32
〈표 15〉 하위 10개의 Attention 을 갖는 형태소	33

그림 목차

[그림 1] 펼쳐 놓은 순환신경망 구조	4
[그림 2] RNN 을 이용한 분류기 모델	6
[그림 3] LSTM Cell 의 구조	7
[그림 4] LSTM Classifier	10
[그림 5] Bi-directional LSTM Classifier	11
[그림 6] Attention-based LSTM Classifier	13
[그림 7] 네이버 리뷰 데이터의 예시	20
[그림 8] Rotten Tomato Movie Review 의 긍정적(상단), 부정적(하단) 리뷰 예시	24
[그림 9] 감성 분석 프로세스	25

제 1 장 서 론

자연어 분석(Natural Language Processing)은 인간이 사용하는 언어 현상을 기계적으로 분석하여 컴퓨터가 이해할 수 있는 방식으로 변환한 뒤 이를 가공하여 활용하는 것을 뜻한다. 그 중에서 감성 분석(Sentiment Analysis)은 주어진 글이 가리키는 대상 또는 화자에 대해 어떤 평가를 내리고 있는지에 대해 분석하는 분야로 개개인의 감성과 의견은 연령이나 성별, 직업과 같은 데이터와 달리 숫자로 쉽게 표현되기 어렵기 때문에 자연어 처리 분야에서도 해결하기 어려운 과제로 여겨져 왔다. 오피니언 마이닝(Opinion Mining)이라고도 불리는 감성 분석은 다양한 분야에서 활용되고 있다. 예를 들어 2012년 열린 미국 대선 중에는 오바마 캠프에서 부동층을 공략하기 위해 소셜 데이터를 수집하여 개인을 대상으로 맞춤형 선거 전략을 펼쳤고 이러한 활동은 오바마가 대통령으로 당선되는데 주요한 영향을 끼쳤다. 또한 기업들은 뉴스나 블로그, 커뮤니티로부터 시장 현황이나 소비자 트렌드, 브랜드 이미지에 대한 추적의 필요성이 늘어나고 있다. 특히 트위터와 페이스북을 중심으로 등장한 소셜 미디어의 등장은 광고와 마케팅의 영역에서 감성 분석의 수요를 높이는데 큰 영향을 끼쳤다. [26]

감성 분석의 방법으로 영상처리와 음성인식 분야에서도 각광받고 있는 딥러닝 기반의 모델이 연구되고 있다. 고전적인 DNN(Deep Neural Network) 모델에서는 모든 입력을 독립적으로 인식하기 때문에 과적합과 높은 시간복잡도를 갖는 문제가 제기되었다. 이 문제를 해결하기 위해 미니 배치(mini batch)와 드롭 아웃(drop out)와 같은 기법이 제시되었고 [1] 특히 딥러닝 모델에 행렬 및 벡터 계산에 특화된 GPU 를 이용함으로써 학습 속도를 비약적으로 향상시켜주었다. [2] 또한 합성곱(Convolution) 연산을 이용하여 인접한 입력

데이터끼리 가중치를 공유하는 합성곱 신경망(Convolutional Neural Network, CNN)과 시퀀스로 입력이 들어오는 경우 재귀적으로 현재 상태를 다음 시간 스텝으로 전달해주면서 각 시간 스텝마다 메모리를 갖게 해 주는 순환신경망(Recurrent Neural Network, RNN)은 자연어 처리 분야에서도 뛰어난 성능을 보여왔다. [3] [4]

자연어 처리에 기계 학습 방법을 적용하기 위해서는 주어진 텍스트로부터 특징을 추출하는 과정(feature extraction)이 필요하다. 특징이란 관찰 대상으로부터 얻어낸 구별 가능하고 측정할 수 있는 속성을 말한다. 자연어 처리에서는 주로 수학적으로 다루기 편한 벡터(feature vector)의 형태로 특징을 추출하여 기계학습에 사용한다. [16]

문서를 어떤 단위로 나누어 특징을 추출하는가에 따라 자연어 처리의 성능이 달라질 수 있다. 일반적으로 단어 단위로 문서를 분할(Segmentation)하여 특징을 추출하는 방법을 사용하지만 최근 연구에서는 글자 단위로 문서를 분할하는 방법을 이용하여 기계번역 등에서 좋은 성능을 보이고 있다. [17] 또한 영어가 아닌 아랍어나 체코어, 핀란드어와 같이 형태론적인 특성이 풍부한 언어의 경우 단어로부터 형태소를 추출하여 특징 추출에 활용하는 방식을 이용하여 성능을 향상시킬 수 있다. [18]

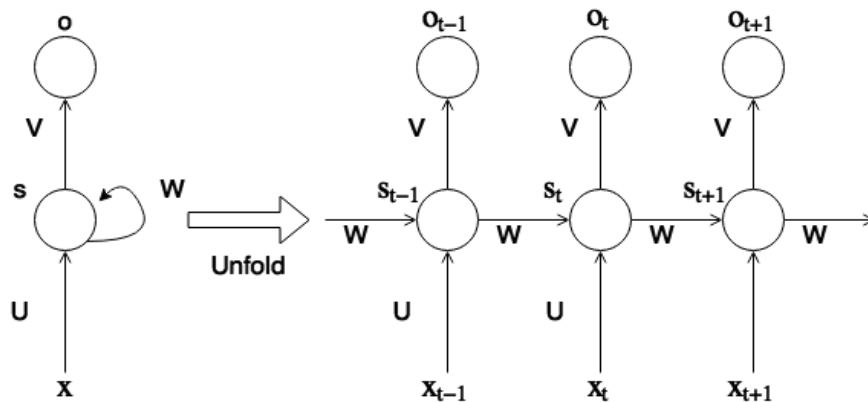
본 연구에서는 순환신경망을 중심으로 한국어로 된 영화 리뷰 데이터를 기반으로 글의 긍정적 또는 부정적 평가를 효과적으로 분류하는 방법에 대한 내용을 다룬다. 한글로 작성된 자연어에 대해서 특징을 추출하는 방법으로 네 가지 방법을 비교해 보았다. 첫째로 글자를 자모 단위로 나누어 문서로부터 특징을 추출하여 RNN 모델의 입력으로 사용하는 방법이다. 둘째로 문장을 글자 단위로 나누어 입력에 사용할 특징을 구성하는 방법을 사용한다. 셋째로는 일반적으로 사용하는 방법인 문서를 단어 단위로 나누어 특징 벡터를 구성해 본다. 마지막으로 단어를 각각 형태소(morpheme)로 변환하여 특징으로

사용하는 방법을 적용한다. 자연어로 부터 위에서 제시한 네 가지 방법을 이용하여 얻은 특징 벡터는 순환신경망 모델의 변형인 LSTM, Bi-directional LSTM 과 Attention Mechanism 의 입력으로 사용하여 모델을 훈련시키고 성능을 평가해 보았다. 또한 Attention Mechanism 을 적용한 모델로부터 문서에 대한 Attention Vector 를 계산하여 Attention 이 갖는 의미를 분석해 보았다.

제 2 장 관련 연구

2.1. 순환신경망(Recurrent Neural Network, RNN)

자연어는 시간의 흐름에 따라 순차적으로 정보들이 배열되어 있다. 고전적인 신경망 모델에서는 모든 입력과 출력이 각각 독립으로 가정하고 있지만 자연어와 같은 실제 데이터는 데이터 간에 관계를 갖고 있다. 예를 들어 어떤 문장이 주어졌을 때 문장 내의 단어의 의미는 단순히 단어의 사전적인 의미가 아닌 전후 맥락에 의해 뜻을 해석해야 경우가 많다. 순환신경망(Recurrent Neural Network, RNN) [4] 은 자연어와 같이 순차적인 정보를 처리하는데에 특화된 뉴럴 네트워크 모델이다. 순환신경망은 이름 그대로 재귀적으로 현재 노드의 상태를 다음 노드에 전달해주면서 동일한 태스크를 한 시퀀스의 모든 요소마다 실행한다. 따라서 순환신경망의 은닉 상태(hidden state)는 현재까지 계산된 결과에 대한 "메모리"에 해당한다고도 볼 수 있다.



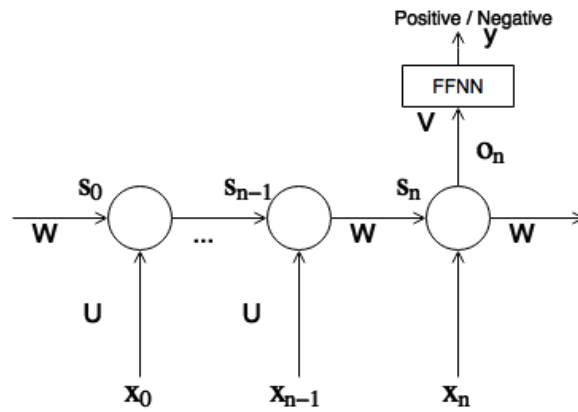
[그림 1] 펼쳐 놓은 순환신경망 구조

[그림 1]은 순환신경망을 펼쳐 놓은(unfold) 구조이다. 여기서 x_t 는 시간 스텝(timestep) t 에서의 입력이고 s_t 는 스텝 t 에서의 은닉 상태, o_t 는 스텝 t 에서의 출력이다. s_t 와 o_t 는 순환 가중치(recurrent weight) W, U, V 그리고 bias b_s, b_o 에 대해 다음과 같이 정의된다.

$$s_t = f(Ux_t + Ws_{t-1} + b_s) \quad (1)$$

$$o_t = Vs_t + b_o \quad (2)$$

여기서 f 는 비선형 활성화함수(activation function)로 주로 \tanh 또는 sigmoid 함수가 쓰인다. 여기서 주목할 점은 모든 시간 스텝 이 학습 파라미터 W, U, V 그리고 bias b_s, b_o 를 공유하고 있다는 것이다. 시간 스텝에 따라 파라미터 수가 늘어나지 않는 순환신경망의 특성 때문에 RNN 을 이용하면 학습해야 하는 파라미터 수를 많이 줄여주어 모델의 복잡도를 낮출 뿐만 아니라 입력 시퀀스(sequence) 내의 입력 데이터끼리 문맥적인 의존 관계(contextual dependency)를 갖게 해 준다.



[그림 2] RNN 을 이용한 분류기 모델

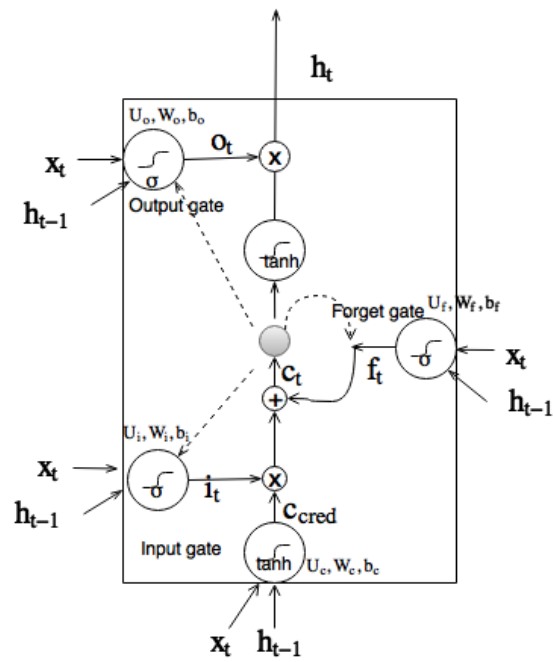
[그림 2]는 RNN 을 이용하여 분류기(Classifier)를 만든 구조를 나타낸다. RNN 에서는 각 시간 스텝의 입력에 대해 은닉 상태와 공유 가중치를 이용하여 계산된 출력이 나오지만 본 연구에 사용할 감정 분류 모델에서는 마지막 시간 스텝에 대한 출력만 사용한다. 시퀀스의 처음부터 은닉 상태는 마지막 시간 스텝에 이르기까지 다음 시간 스텝에 전달되고 학습 파라미터는 모든 시간스텝이 공유하기 있기 때문에 마지막 시간 스텝의 출력을 이용하는 것으로 시퀀스의 형태로 들어오는 입력에 대해 단일한 출력을 내는 것으로 모델링 할 수 있다. RNN 모델의 마지막 시간 스텝의 출력은 FFNN(Feed-Forward Neural Network)의 입력으로 사용된다. FFNN 을 통과하여 계산된 출력값은 loss function 으로 사용하여 학습 및 추론에 사용된다.

순환신경망을 학습할 때는 순차적으로 들어오는 입력에 따라 시간에 따른 오류역전파 기법(Back Propagation Through Time, BPTT)을 사용한다. [5] 순환 신경망은 입력 시퀀스 길이만큼 layer 가 깊어지기 때문에 길이가 긴 시퀀스에 대해 학습을 하게 되면 어떤 시간 상태의 gradient 값이 레이어를 거쳐 전달될수록 0으로 수렴하게 되어 먼 시간 스텝에 이르러서는 값이 거의

전달되지 못하게 되고, 먼 과거 시간 스텝의 상태(state)가 현재 스텝의 학습에 아무런 도움이 되지 못하게 된다는 한계를 갖고 있다. 이렇게 장기 의존성에 대해 학습이 올바르게 수행되지 않는 문제를 Vanishing Gradient 라고 하는데 이후 설명할 LSTM(Long-Short Term Memory) 방법에 의해 효과적으로 해결되었다.

2.2. LSTM(Long-Short Term Memory)

1997년 처음 제안된 LSTM(Long-Short Term Memory)[6]은 RNN(Recurrent Neural Network)에서 학습이 진행될수록 초기 상태의 파라미터의 갱신이 작아지게 되는 Vanishing Gradient 문제를 해결하기 위해 제안된 네트워크이다. 하나의 LSTM 노드는 [그림 3]과 같이 입력 게이트(Input Gate), 망각 게이트(Forget Gate), 그리고 출력 게이트(Output Gate)의 세 개의 게이트로 이루어져 있는 셀(Cell)로 구성되어 있다.



[그림 3] LSTM Cell 의 구조

입력 게이트에서는 새로 입력받은 데이터와 이전에 입력받은 데이터를 같이 사용할지에 대해 결정한다. 망각 게이트에서는 이전 데이터에 대한 학습 데이터를 유지할지 버릴지에 대한 여부를 결정한다. 그리고 출력 게이트에서는 현재까지 계산된 부분을 어느정도 출력할지를 결정한다. 셀 내부에서 일어나는 연산은 다음과 같다.

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) \quad (4)$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) \quad (5)$$

$$c_{cand} = \tanh(U_c x_t + W_c h_{t-1} + b_c) \quad (6)$$

$$c_t = c_{t-1} \cdot f_t + c_{cand} \cdot i_t \quad (7)$$

$$h_t = \tanh(c_t) \cdot o_t \quad (8)$$

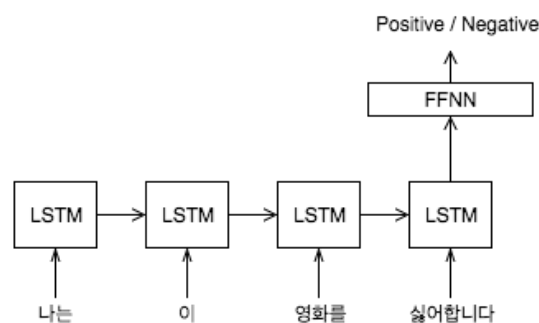
i_t, f_t, o_t 는 각각 입력 게이트, 망각 게이트, 출력 게이트를 가리키며 각각은 모두 시그모이드 함수를 거쳐 모두 0 이상 1 이하의 실수로 정의된다. c_{cand}, c_t, h_t 는 각각 셀 상태 후보, 셀 상태(the cell state), 출력(output)을 나타낸다. 현재 시간 스텝의 셀 상태(c_t)를 계산할 때 망각게이트의 정도에 따라 이전의 셀 상태(c_{t-1})를 유지하고 입력 게이트의 정도에 따라 새로 계산한 셀 상태(c_{cand})를 받아들인다. 그리고 여기에 \tanh 의 활성화 함수(activation function)를 적용한 뒤 출력 게이트 만큼을 셀의 출력(h_t)으로 사용한다. LSTM은 셀 상태의 개념을 도입하고 이를 입력, 망각, 출력 게이트를 이용하여 현재의 셀 상태를 다음 셀로 안정적으로 전달함으로써 기존 순환신경망이 갖는 vanishing gradient 문제를 회피하였다.

LSTM은 시간을 거쳐 여러 변형이 생겼는데 대표적으로 Cho, et, al. (2014)에서 제안된 GRU(Gated Recurrent Unit)[7]가 있다. GRU는 입력 게이트와 망각 게이트를 하나의 업데이트 게이트(Update Gate)로 통일시키고 출력에 계산하는 비선형 함수를 제거하는 등 기존의 LSTM 모델을 단순화시켜 빠른 학습을 가능하게 하였다. Greff, et. al.(2015) [8]에서는 GRU를 포함하여 8가지의 LSTM의 변형 모델에 대해서 성능을 비교하였는데 특별하게 다른 모델보다

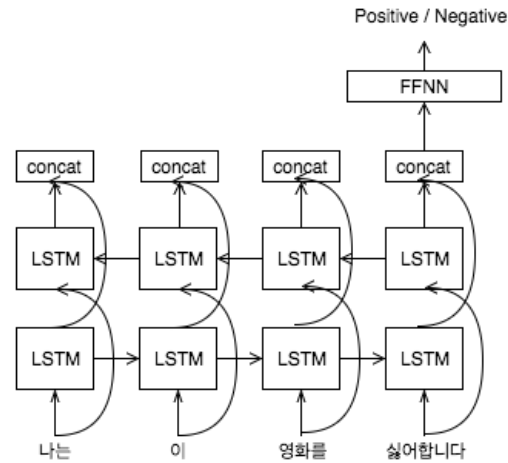
뛰어난 성능을 보이는 모델은 없었다고 결론을 내렸다. 이것을 근거삼아 본 연구에서는 기본적인 LSTM 을 이용하는 것만으로도 높은 성능을 보이는 것을 기대하여 LSTM 실험에 사용하였다.

2.3. Bi-directional LSTM

Bi-directional LSTM 은 기존 LSTM 모델과 같은 크기이지만 방향이 반대인 LSTM Layer 를 수직으로 쌓아 양방향으로 네트워크를 순환하도록 하는 방법이다. LSTM 을 포함한 RNN 은 데이터의 이전 상태 정보를 메모리의 형태로 저장할 수 있는데 Bi-directional LSTM[9]은 여기서 더 나아가 어떤 시간 스텝에서 과거의 상태 정보 뿐만 아니라 미래의 상태 정보까지 저장하여 학습에 사용한다. Bi-directional LSTM 의 출력은 각기 다른 방향으로 흐르는 두 층의 LSTM 의 출력을 이어붙여(concatenate) 사용한다. Bi-directional LSTM 의 출력은 기존 LSTM 상태의 두 배의 크기를 사용하기 때문에 실제로 모델을 구현할 때 이 점을 주의하여 벡터 크기를 설정해 주어야 한다.



[그림 4] LSTM Classifier



[그림 5] Bi-directional LSTM Classifier

2.4. Attention Mechanism

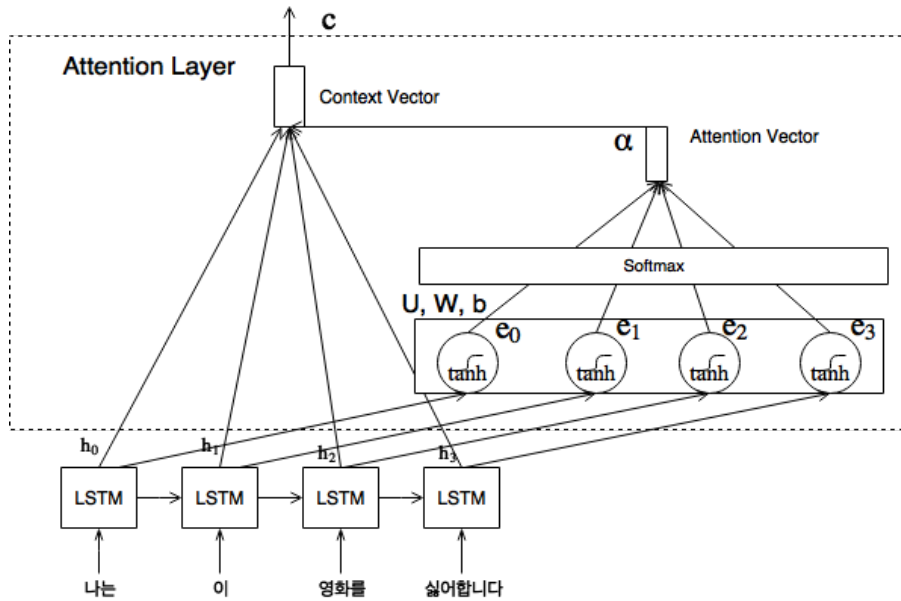
LSTM 을 포함한 기본적인 RNN 모델은 길이에 관계 없이 고정된 크기로 입력 데이터를 인코딩하는데 입력 데이터마다 길이의 차이가 클 경우 성능이 떨어진다는 문제가 발생한다. 여기서 RNN 의 각 시간 스텝의 출력에 Attention Mechanism 을 적용함으로써 이 문제를 해결할 수 있다. attention mechanism 에서는 모든 시간 스텝의 출력을 입력으로 받아 가중치 합(weighted sum) 연산을 수행함으로써 각 시간 스텝에 대한 주목의 정도(attention)를 얻을 수 있다. Attention Layer 에서 수행하는 연산은 다음과 같다.

$$e_t = U^T \tanh(W o_t + b) \quad (9)$$

$$a_t = \text{softmax}(e_t) = \frac{\exp(e_t)}{\sum_k^N \exp(e_k)} \quad (10)$$

$$c_t = \sum_t^N a_t o_t \quad (11)$$

e_t 는 시간 스텝 t 에서 LSTM의 출력 o_t 를 입력받아 학습 파라미터 W , U , b 를 이용하여 계산한 가중치 벡터로 여기에 softmax 연산을 하여 attention vector a_t 를 얻는다. attention vector는 context vector c 를 구할 때 각 시간 스텝 t 의 출력 o_t 에 대한 가중치 벡터로 사용된다. context vector는 attention layer의 출력으로 앞서 설명한 분류기 모델과 같이 Feed-forward neural network의 입력으로 사용된다. c 를 구하는 식에서 알 수 있듯 attention vector는 시퀀스 전체에 대해 시간 스텝 t 가 얼마나 더 주목(attention)받는지에 대한 정보를 포함하고 있기 때문에 a_t 를 분석함으로써 어떤 단어 또는 말뭉치가 문서가 판단 또는 감정적인 표현을 하는 데 있어서 상대적으로 더 영향을 미치는지 파악할 수 있다.



[그림 6] Attention-based LSTM Classifier

Attention Mechanism 을 응용하여 다양한 모델이 제안되었다. Yang et al., 2016에서는 Attention Layer 를 복층으로 나누어 Word-Level 과 Sentence-Level 를 나누어 많은 문장이 포함된 문서에 대해 좋은 성능을 보이게 하는 HAM(Hierarchical Attention Networks)[11] 을 제시하였고, Wang et al., 2016에서는 Attention Layer 연산 과정에서 얻어지는 Aspect Embedding 을 단어 표현(Word Representation) 전에 붙여 성능을 향상시킨 ATAE-LSTM(Attention-based LSTM with Aspect Embedding)을 제안하였다. Attention Mechanism 은 기계 번역 분야에서 뛰어난 성능을 보이며 이를 응용하여 Google 에서는 Sequence to Sequence 과 같은 모델을 제안하였다.[13] Show, Attend and Tell [14] 에서는 2D 이미지를 입력으로 주어 Convolutional Neural Network 과 Attention Mechanism 을 융합하여 특정 토픽이 가리키는 위치를 2D 이미지 상의 픽셀의 분포로 시각화하였다.

제 3 장 한국어의 RNN 을 이용한 감성 분석

3.1. 한국어의 벡터 표현 방식

자연어로 구성되어 있는 문서를 RNN 을 포함한 자연어 처리 모델을 이용하여 분석하기 위해서는 텍스트 데이터를 컴퓨터가 이해할 수 있는 형태, 즉 벡터의 형태로 변환하는 과정이 필요하다. 자연어를 벡터로 변환하는 방법으로 주로 문서를 구성 요소로 쪼개(segmentation) 뒤 각각의 구성 요소를 고유한 숫자로 매핑하는 방법을 사용한다. [19] 문서의 구성 요소로는 글자나 단어, 또는 단어에 대해 품사를 분석한 뒤 얻어진 형태소를 사용한다. 기존에는 영어로 된 텍스트에 대해서 단어 단위로 글을 분할하여 자연어 처리에 활용하는 방식을 이용하여 기계 번역 등에서 높은 연구 성과를 보여왔다. 하지만 단어 단위의 표현 방법에는 사전에 없는 단어(out-of-vocabulary)를 올바르게 모델링하지 못한다는 단점과 터키어나 핀란드어와 같은 풍부한 형태(rich morphology)를 갖는 언어에 대해서 언어 모델을 올바르게 생성하지 못한다는 한계를 갖는다. [18] 한글의 경우 자음과 모음, 그리고 받침으로 구성된 자모를 이용하여 하나의 글자를 만드는 언어적 특징을 가지고 있다. 이 특징을 이용하여 문서 내의 글자를 자모 단위로 쪼개어 문서의 구성 요소로 사용할 수 있다. <표 1>은 '재밌다.' 라는 글이 있을 때 이를 자모, 글자, 단어, 그리고 3장 2절에서 설명할 형태소 단위로 나누어 각 구성 요소로 구성된 배열(Array)로 표현한 것이다.

〈표 1〉 ‘재밋다.’라는 문서를 다양한 문서의 구성 요소로 쪼갠 결과

	원문	문서를 분할한 결과
자모	재밋다.	[ㅈ, ㅞ, ㅁ, ㅣ, ㅍ, ㅊ, ㄷ, ㅏ, .]
글자	재밋다.	[재, 밋, 다, .]
단어	재밋다.	[재밋다.]
형태소	재밋다.	[재미/Noun, 있다/Verb, ./Punctuation]

문서를 어떠한 방식으로 단위화하는가에 따라 사전의 크기와 시퀀스의 길이가 달라진다. 사전의 크기는 모델을 훈련 및 추론할 때 필요한 메모리의 크기에 선형적으로 비례하고 [20], 시퀀스의 길이가 길어지면 RNN 모델의 노드 수가 증가하게 된다.

〈표 2〉 자모 단위로 분할한 글자에 매핑된 사전

ㄱ	ㄴ	ㄷ	ㅍ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㅣ	.
1	2	3	4	5	6	7	8	9	10	11

〈표 3〉 문서를 자모 단위로 나눈 뒤, 매핑 후 임베딩하는 과정

원문	연기가 좋았다.
단위화	[ㅇ,ㅋ,ㄴ,ㄱ,ㅣ,ㄱ,ㅏ,ㅓ,ㅓ,ㅇ,ㅏ,ㅓ,ㅓ,ㅓ,ㅣ,.]
매핑	[4, 8, 2, 1, 10, 1, 7, 6, 9, 5, 1, 4, 3, 7, 11]
임베딩	[[0, 0.0231, ..., 0.5], [0.672,0,....,0.001], ..., [0.22,0.665,....,0.8]

단순히 문서의 구성 요소에 숫자를 매핑하는 것 만으로는 구성 요소간의 차이가 대수적으로 크게 나타나지 않기 때문에 자연어 처리에서 충분히 좋은 성능을 얻기 위해서는 매핑된 숫자를 다시 다차원의 벡터로 변환할 필요가 있다. 본 연구에서는 임베딩으로 인한 성능의 변화를 통제하기 위해 자모, 글자, 단어, 형태소로 표현된 모든 경우에 대해서 -1에서 1사이의 값을 갖는 연속균등분포(continuous uniform distribution)를 갖는 분산 표현(distributed representation)의 방식으로 임베딩을 수행하여 모델의 입력 데이터로 사용하였다. 문서를 나누는 단위가 독립적인 뜻을 가질 경우에는 비슷한 단어들을 가까운 위치에 배치하게 하는 Word2vec 과 같은 방식이 제안되지만 [15] 이러한 접근은 독립적으로는 아무런 의미를 갖지 않는 자모나 글자의 단위로 문서를 표현할 경우에는 적용하기 어렵기 때문에 제외하였다.

3.2. 형태소 분석

형태소는 의미를 갖는 가장 작은 말의 단위를 일컫는 말이다. 형태소를 분석한다는 것은 불규칙 활용이나 축약, 탈락 현상이 일어난 글의 어절에 대해서 그것의 원형인 형태소를 찾는다는 것을 의미한다. 한국어의 경우 ‘영화’라는 명사가 있을 때 뒤에 이어지는 내용에 따라서 ‘영화가’, ‘영화는’, ‘영화를’, ‘영화관’, ‘영화이다’ 등 접사나 기능어가 붙거나 용언화 등 다양한 활용(conjugation)을 갖기 때문에 단순히 문장을 단어 또는 어절로 분류하는 것으로는 좋은 성능을 얻지 못한다.

본 연구에서는 형태소 분석을 위해 트위터에서 에서 만든 형태소 분석기인 twitter-korean-text[27] 를 사용한다. twitter-korean-text에서는 품사를 19개로 나누며 그것은 각각 명사(Noun), 동사(Verb), 형용사(Adjective), 관형사(Determiner), 부사(Adverb), 접속사(Conjunction), 감탄사(Exclamation), 조사(Josa), 선어말어미(PreEomi), 어미(Eomi), 접미사(Suffix), 구두점(Punctuation), 외국어(Foreign), 알파벳(Alpha), 숫자(Number), 미등록어(Unknown), 초성어(KoreanParticle), 해쉬태그(Hashtag), 스크린네임(ScreenName)과 같다.

〈표 4〉 잘못된 형태소 분석의 예

원문	재밋기만한데;;왜이렇게욕을하생휴ㅠㅠ
형태소 분석	('재밋다', 'Adjective') ('말다', 'Verb') (';;', 'Punctuation') ('왜', 'Noun') ('이렇게', 'Adverb') ('욕', 'Noun') ('을', 'Josa') ('하생', 'Noun') ('휴', 'Exclamation') ('ㅠㅠ', 'KoreanParticle')

형태소 분석은 다른 데이터셋을 통해 사전에 학습된 단어를 기반으로 품사를 붙이기 때문에 신조어나 잘 사용하지 않는 희귀한 단어가 나오거나, 맞춤법이 틀린 문서에 대해서 잘못된 의미의 형태소로 변환될 위험이 있다. [표 2]를 보면 ‘-만한데’ 라는 단어가 ‘말다’라는 동사의 형태로, ‘하생’이라는 단어를 명사로 잘못 분석한 것을 확인할 수 있다.

제 4 장 실험 및 결과

4.1. 실험 데이터

본 연구에서는 실험에 사용할 데이터셋으로 네이버 영화 리뷰 데이터[28]와 Rotten Tomato Movie Review[22]를 이용한다. 네이버 영화 리뷰 데이터는 네이버 영화의 네티즌 평점 · 140자평 서비스[29]를 웹 크롤링 한 후 가공한 것이다. 리뷰 데이터는 id 와 document, label 로 구성되어 있다. id 는 각 리뷰 데이터에 대해 부여된 고유한 번호이다. document 는 영화평에 대한 140자 이하의 텍스트로 구성되어 있으며 한글을 포함한 영어, 숫자, 특수문자를 포함하고 있다. label 은 사용자가 영화에 대해 숫자로 남긴 평가이다. 네이버 영화 서비스에서는 평점을 1이상 10 이하의 정수로 남길 수 있는데 실험에서는 5~8점의 중립적인 평가를 남긴 데이터를 제외하고 1~4점으로 부여한 리뷰는 부정적(Negative), 9~10점의 평가를 내린 리뷰는 긍정적(Positive)로 분류하여 부정적인 평가의 경우 0, 긍정적인 평가를 내린 경우 1로 라벨링하였다. 네이버 영화 서비스로 부터 평점이 1~4점, 그리고 9~10점으로 매겨진 20만개의 리뷰 데이터를 추출한 뒤 15만개는 학습에 사용되고 5만개는 검증에 사용하였다. 추출한 15만개의 학습 데이터와 5만개의 검증 데이터는 모두 같은 비율로 긍정적 평가와 부정적 평가를 분포시켜 특정 라벨로 데이터가 쏠리는 현상을 방지하였다.

id	document	label
6270596	굳 ㅋ	1
9274899	GDNTOPCLASSINTHECLUB	0
8544678	뭐야 이 평점들은.... 나쁘진 않지만 10점 짜리는 더더욱 아니잖아	0
6825595	지루하지는 않은데 완전 막장임... 돈주고 보기에는....	0
6723715	3D 만 아니어도 별 다섯 개 줬을텐데.. 왜 3D 로 나와서 제 심기를 불편하게 하죠??	0
7898805	음악이 주가 된, 최고의 음악영화	1
6315043	진정한 쓰레기	0
6097171	마치 미국애니에서 튀어나온듯한 창의력없는 로봇디자인부터가,고개를 젓게한다	0
8932678	갈수록 개판되가는 중국영화 유치하고 내용없음 품잡다 끝남 말도안되는 무기에 유치한 cg 납무 아 그림다 동사서독같은 영화가 이진 3류아류작이다	0
6242223	이별의 아픔뒤에 찾아오는 새로운 인연의 기쁨 But, 모든 사람이 그렇지는 않네..	1
7462111	괜찮네요오랜만포켓몬스터잼맞어요1	

[그림 7] 네이버 리뷰 데이터의 예시

실험에 사용할 리뷰 데이터의 문서는 전처리를 거치지 않은 실제 사용자가 남긴 글과 평가를 사용하기 때문에 맞춤법이 맞지 않거나 은어 또는 신조어를 사용하는 등 문법이 맞지 않는 문장이 다수 포함되어 있다. <표 5>은 네이버 리뷰 데이터에 포함된 비문의 유형과 각각의 예시를 보여준다. 실험에 사용할 데이터는 <표 5>과 같은 비문을 모두 포함한 문서로 구성되어 있으며 특별한 전처리는 수행하지 않았다.

〈표 5〉 네이버 리뷰 데이터에 포함된 비문과 예시

	예시 텍스트
초성체	팬이긴한데 이걸 안봤네요ㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋㅋ몰라 나도 안보면 후회ㅠㅠ...
이모티콘	음악과 인간과 그 시대상이 모두 적절히 어우러진 감동적인 영화였어요 :) -.,- 히접스런 영화,스토리도 시대착오적인..3류 코미디
띄어쓰기 오류	괜찮네요오랜만포켓몬스터잼밧어요 전기톱은못들고다니는데 엔진톱이겠쵸
틀린 맞춤법	완전 잼없음..
신조어의 사용	하나같이 발연기에 있는 폼 없는 폼 다잡구 있네 견자단나오는 영화중에 제일 노잼

한국어를 어떤 단위로 벡터로 표현하는가에 따라서 사전의 크기와 최대 시퀀스의 크기가 달라질 수 있다. 〈표 6〉는 네이버 리뷰 데이터에 대해서 한국어 표현 방식에 따른 사전의 크기와 최대 시퀀스 크기를 보여준다. 최대 시퀀스의 길이가 작아질수록 사전의 크기가 급격히 커지는 경향을 확인할 수 있다. 사전의 크기에 따라 모델을 훈련 및 추론할 때 필요한 메모리의 크기가 결정되고, 시퀀스의 길이에 따라 RNN 모델의 노드 수가 결정된다. 문서 단위간 차이를 만들기 위해 위해 본 실험에서는 매핑된 입력 데이터에 -1에서

1사이의 값을 갖는 100차원의 연속균등분포로 임베딩하여 모델의 입력으로 사용하였다.

〈표 6〉 네이버 영화 리뷰 데이터의 표현 방식에 따른
사전의 크기와 최대 시퀀스 크기

	사전의 크기	최대 시퀀스 길이
자모	647	418
글자	3226	158
단어	450533	42
형태소	70338	95

Rotten Tomato Movie Review[22] 는 Rotten Tomato 서비스로부터 추출한 영화에 대한 5311개의 긍정적 평가와 5311개의 부정적 평가를 나타낸 영화 리뷰 데이터이다. 이 중 긍정적 평가를 나타낸 533개의 리뷰와 부정적 평가를 나타낸 533의 리뷰를 검증 데이터로 사용하고 남은 9556개의 리뷰 데이터를 훈련에 사용하였다. Rotten Tomato Movie Review 의 모든 리뷰 문서는 영어로 작성되어 있으며 한국어로 작성된 네이버 리뷰 데이터와 언어적인 차이를 보기 위해 동일한 모델로 실험에 사용하였다. Rotten Tomato Movie Review 데이터를 형태소로 변환할 때에는 NLTK[30]의 Tokenizer 를 이용하였다.

if you sometimes like to go to the movies to have fun , wasabi is a good place to start .

the film provides some great insight into the neurotic mindset of all comics -- even those who have reached the absolute top of the game .

(wendigo is) why we go to the cinema : to be fed through the eye , the heart , the mind .

one of the greatest family-oriented , fantasy-adventure movies ever .

simplistic , silly and tedious .

it's so laddish and juvenile , only teenage boys could possibly find it funny .

exploitative and largely devoid of the depth or sophistication that would make watching such a graphic treatment of the crimes bearable .

[garbus] discards the potential for pathological study , exhuming instead , the skewed melodrama of the circumstantial situation .

[그림 8] Rotten Tomato Movie Review 의

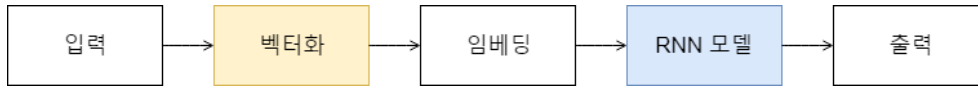
긍정적(상단), 부정적(하단) 리뷰 예시

〈표 7〉 Rotten Tomato Movie Review 의 표현 방식에 따른
사전의 크기와 최대 시퀀스 크기

	사전의 크기	최대 시퀀스 길이
글자	90	269
단어	21389	60
형태소	25522	62

4.2. 실험 방법

[그림 9]는 본 연구에서 실험에 사용할 감성 분석 모델의 처리 과정을 나타낸다. 먼저 네이버 영화 리뷰 데이터의 document 를 입력으로 받는다. 이후 벡터화 과정에서는 자모, 글자, 단어, 그리고 형태소의 단위로 자연어로 된 문서를 벡터화 한 뒤에 임베딩 과정을 수행하여 RNN 모델의 입력 데이터로 사용한다. RNN 모델로는 [그림 4]와 같이 LSTM 을 시퀀스 길이만큼 연결한 모델, [그림 5]와 같이 반대 방향으로 흐르는 두 LSTM 을 수직으로 연결한 Bi-directional LSTM 모델, 그리고 LSTM 과 Bi-directional LSTM 에 각각 Attention Layer 를 추가한 Attention Mechanism 모델, 그리고 대조군으로 순환신경망 모델이 아닌 분류기 모델인 Naïve Bayes Classifier[23] 모델을 사용한다.



[그림 9] 감성 분석 프로세스

실험은 크게 훈련(train)과 검증(test)으로 나눌 수 있다. 훈련 과정에서는 과적합(Over-fitting)을 방지하기 위해 LSTM 셀과 Attention Layer의 출력에 0.5의 drop-out 을 적용하였다. 각 모델에 연결된 Feed-forward Neural Network의 출력값은 훈련 데이터의 라벨 값과 차이를 이용하여 sigmoid 를 취한 뒤 Cross-Entropy 를 이용하여 loss function 으로 사용하였다. FFNN의 출력을 x , 라벨을 y 라고 할 때 loss function 을 구하는 식은 다음과 같다.

$$L(x, y) = -\log(\sigma(x)) + (1 - y) * (-\log(1 - \sigma(x))) \quad (11)$$

여기서 σ 는 sigmoid 함수, 즉 $\sigma(x) = 1/(1 + \exp(-x))$ 를 가리킨다. 가중치의 갱신은 loss function 의 값을 최소화하는 방향으로 시간에 따른 오류역전파(BPTT) 알고리즘을 통해 이루어진다. 여기서 파라미터가 local minima 에 빠지는 것을 방지하고 학습을 빠르게 하기 위해 Adam(Adaptive Moment Estimation) optimization 기법을 사용하였다. 추론은 모델의 출력에 sigmoid 함수를 통과시켜 0.5 이상이면 긍정(Positive), 0.5 이하이면 부정(Negative)로 판단한다. 추론을 통해 정확도(Accuracy)를 계산하고 이를 감정 분류기의 성능의 지표로 사용한다. <표 8>는 머신 러닝의 성능을 평가하는 지표로 자주 사용되는 혼동 행렬(Confusion matrix)을 나타낸 표이다. [21] 만약 어떤 문서가 1로 라벨링 되어 있고, 이 문서에 대한 추론의 결과가 긍정일 경우

True Positive, 1로 라벨링된 문서에 대해 부정으로 판단한 경우 False Negative, 0으로 라벨링 된 데이터에 대해 긍정으로 판단한 경우 False Positive, 0으로 라벨링 된 데이터를 부정으로 판단한 경우 True Negative 로 분류한다. 모든 검증 데이터에 대해 True Positive, False Positive, True Negative, False Negative 의 갯수를 각각 센 뒤 (12)의 식을 이용하여 정확도(Accuracy)를 계산한다.

〈표 8〉 이진 분류에서 사용하는 혼동 행렬

	긍정적인 리뷰	부정적인 리뷰
긍정적으로 판단	TP(True Positive)	FP(False Positive)
부정적으로 판단	FN(False Negative)	TN(True Negative)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (12)$$

훈련은 한 epoch 에서 훈련 과정과 검증 과정을 반복하며 검증 과정에서의 정확도가 더이상 높아지지 않을 때까지 반복한다. epoch 이 완료된 이후 훈련이 완료된 모델을 이용하여 임의의 텍스트 입력에 대해 attention 을 얻어 시각화와 정성적 분석에 사용한다.

Naïve Bayes Classifier 는 텍스트 분류에 오랫동안 이용된 방법으로 특성들 사이의 독립을 가정하는 베이즈 정리를 이용한 분류 모델이다. (13) 식은 Naïve Bayes 확률 모델과 결정 규칙을 조합한 것으로 클래스 C_k 에 대해 최대 확률을 갖는 클래스 k 를 찾는 식을 나타낸다.

$$\hat{y} = \operatorname{argmax}_k p(C_k) \prod_i p(x_i | C_k) \quad (13)$$

4.3. 실험 결과

〈표 9〉는 네이버 영화 리뷰 데이터에 대해서 한국어 표현 방식에 따라, 그리고 RNN 모델에 따른 추론 정확도를 나타낸 것이다. 모든 모델에서 문서를 단어 단위로 나누어 벡터로 표현하였을 때 가장 성능이 낮게 나왔다. 자모 단위와 글자 단위로 문서를 표현하였을 때는 LSTM 에 비해 Bi-directional LSTM 의 모델을 사용할 경우 0.38% 이상 좋은 성능을 나타내었지만 LSTM 와 Bi-directional LSTM 모델 모두 Attention Mechanism 을 적용한 모델이 0.46% 이상의 높은 성능을 나타내었으며 형태소 단위로 글자를 나타냈을 때는 큰 차이를 보이지 않았다. Naïve Bayes 의 경우 자모 표현에서 가장 낮은 61.85%를 보였으며 글자나 단어, 형태소의 표현으로 변경할 수록 성능이 높아지는 것을 확인할 수 있었다. Rotten Tomato Movie Review 의 경우 모든 모델에서 단어 단위로 문서를 표현한 경우가 형태소 단위로 표현한 경우보다 성능이 향상되는 것을 확인할 수 있었다.

〈표 9〉 네이버 영화 리뷰 데이터의
RNN 및 Naïve Bayes 모델의 정확도(%)

	Naïve Bayes	LSTM	BLSTM	Attention + LSTM	Attention + BLSTM
자모	61.85	85.21	85.34	86	85.8
글자	76.98	84.78	85.16	86	86.1
단어	80.42	77.99	77.78	76.4	76.7
형태소	83.5	85.88	85.89	85.7	86.2

〈표 10〉 Rotten Tomato Movie Review 의
RNN 및 Naïve Bayes 모델의 정확도(%)

	Naïve Bayes	LSTM	BLSTM	Attention + LSTM	Attention + BLSTM
글자	54.50	60.32	63.88	71.1	73.7
단어	76.26	78.05	78.4	76.8	78.5
형태소	71.47	73.82	73.35	74.7	75.1

Naïve Bayes 의 경우 네이버 리뷰 데이터와 Rotten Tomato Movie Review 모두에서 자모나 글자 단위로 문서를 표현한 경우 성능이 저하되는 것을 확인할 수 있는데 이는 Naïve Bayes 이 모든 입력 원소 간에 독립을 가정하는 베イズ 정리의 한계라고 볼 수 있다. 글자나 자모는 각각에 대해서는

어떠한 의미를 갖지 않기 때문에 시퀀스를 인식하지 못하고 독립이 되는 순간 문서를 올바르게 분류하지 못 하게 된다.

네이버 영화 리뷰 데이터에서 단어 단위로 한국어를 표현할 때 성능이 저하된 이유는 위에서 언급한 단어의 활용(conjugation)으로 인해 같은 의미를 갖는 단어들이 모두 다른 의미로 해석이 되어 입력 데이터의 특징 정보가 파편화되었기 때문으로 보인다. <표 11>는네이버 영화 리뷰 데이터의 미등록어 비율을 나타낸 표이다. 검증 데이터에서 훈련 데이터에 나타나지 않은 단어 또는 글자로 OOV(Out-Of-Vocabulary)라고도 불린다. 다른 문서 표현 방식은 모두 1% 아래로 미등록어가 나타나는데 비해 단어 단위로 문서를 표현한 경우 검증 데이터의 문서 전체에서 25.21%의 단어가 올바르게 인식되지 못한다는 것을 뜻한다. Rotten Tomato Movie Review 의 경우 단어 표현에 대해 5.3% 정도의 손실이 발생하고 있고 단어를 형태소로 변환할 경우 오히려 0.9% 증가하는 것을 확인할 수 있다. 네이버 영화 리뷰 데이터와 달리 Rotten Tomato Movie Review 에서 단어 표현에 비해 형태소 표현에 대해 성능이 낮아지는 것은 미등록어 비율의 증가가 원인으로 보인다. 미등록어의 비율이 높다는 것은 모델을 훈련 데이터로부터 학습한 모델을 검증 데이터에 올바르게 적용하지 못한다는 것을 뜻하고 이는 곧 성능 저하로 이어지게 된다.

〈표 11〉 네이버 영화 리뷰 데이터의 미등록어 비율

	훈련 데이터 단위 수	검증 데이터 단위 수	검증 데이터의 미등록어 단위 수	미등록어 비율
자모	528	336	73	0.00002
글자	5285609	1767847	258	0.00163
단어	1137741	380475	95932	0.25213
형태소	2194536	734551	7842	0.01067

〈표 12〉 Rotten Tomato Movie Review 의 미등록어 비율

	훈련 데이터 단위 수	검증 데이터 단위 수	검증 데이터의 미등록어 단위 수	미등록어 비율
글자	1103586	124210	0	0
단어	201410	22615	1207	0.053
형태소	206241	23100	1496	0.064

문서를 임베딩할 때 글자나 자모 단위로 쪼개는 경우 미등록어의 비율은 줄어들지만 시퀀스 길이가 3배 이상 증가하게 된다. 이는 곧 훈련이 어려워지는 현상을 초래하는데 (Graves, 2013)[24]에서는 Character-level RNN 으로 언어 모델(language model)을 구성하였을 때 Penn Treebank Experiments 와 같은 예측 모델에서 Word-level RNN 에 비해 성능이 저하되는 현상을 보여준다. 다만 실험 결과에서 확인할 수 있듯 RNN 모델에 Attention Mechanism 을 적용할 경우 글자나 자모 단위에서도 충분히 좋은 성능을 얻어내는 것을 알 수

있는데 (Luong, 2015)[25] 에서는 Attention mechanism 을 적용한 경우 그렇지 않은 모델에 비해 문장의 길이가 길어져도 성능이 크게 저하되지 않고 기계 번역 분야에서 좋은 성능을 보이는 것을 보여준다. 글자 표현에서 Rotten Tomato Movie Review 에 비해 네이버 영화 리뷰 데이터가 크게 성능이 떨어지지 않는 것은 26개의 알파벳으로 표현되는 영어에 비해 11,172개의 조합을 갖는 한글 각각의 글자가 특징(feature)으로서 기능한다는 것을 보여준다.

〈표 13〉은 검증 데이터에서 형태소 단위로 문서를 나눈 뒤 정량적 평가에서 가장 좋은 성능을 보였던 Bi-directional LSTM 에 Attention Mechanism 을 적용한 모델을 이용하여 Attention 이 입력 데이터에 대해 작용하는 정도를 나타낸 것이다. 각 입력 데이터에 해당하는 Attention vector 를 구한 뒤 문서 단위로 normalize 하여 형태소 별로 attention 의 합을 구하였다. 〈표 13〉에서 확인할 수 있듯 입력 문서로부터 ‘발연기’, ‘싫어지다’, ‘따뜻하다’ 와 같은 형태소가 활성화 되는 것을 확인할 수 있었다.

〈표 13〉 입력 문서에 대한 Attention 벡터의 활성화에 대한 시각화

원문	발연기 도저히 못보겠다 진짜 이렇게 연기를 못할거라곤 상상도 못했네
	발연기 도저히 못 보다 진짜 이렇게 연기 를 못 하다 상상 도 못 하다
원문	평점1점도 주기싫어지는 영화 배우나 감독이라는 사람이나 영화관에서 안봐으면 한다
	평점 1 점도 주기 싫어지다 영화 배우 나 감독 이라는 사람 이나 영화 관 에서 안 보다 하다 그리고 평 점 알바생 들 너무 티난다
원문	지금까지 본 영화중 마음이 가장 따뜻해지는 영화.
	지금 까지 보다 영화 중 마음 이 가장 따뜻하다 영화 !

모델이 긍정적으로 판단한 데이터와 부정적으로 판단한 데이터를 나누어 각각을 Attention 값에 대해 정렬한 결과이다. 하지만 <표 13> 위와 같이 ‘영화’라는 형태소는 긍정적으로 판단한 글과 부정적으로 판단한 글 모두에게 나타나는 것을 확인할 수 있었다. 이렇게 모든 판단에 관여하는 형태소를 배제하기 위해 긍정적으로 판단한 데이터의 Attention 은 더하고, 부정적으로 판단한 데이터의 Attention 은 뺄셈을 적용하였다. <표 14>과 <표 15>은 Attention 의 순서대로 상위 10개의 형태소와 하위 10개의 형태소를 추출한 것이다. ‘재밌다’, ‘좋다’, ‘쓰레기’, ‘최악’과 같은 직관적으로 영화에 대한 판단을 내리는데 주요한 형태소가 나타나는 것을 확인하였다.

<표 14> 상위 10개의 Attention 을 갖는 형태소

형태소	품사	Attention value
재밌다	Adjective	2315.697379
좋다	Adjective	1644.521349
최고	Noun	1389.613749
!	Punctuation	946.5683639
재미있다	Adjective	817.3605112
보다	Verb	761.4358254
감동	Noun	672.1387658
의	Josa	658.8594882
!!	Punctuation	498.2644915
명작	Noun	434.2194712
ㅋㅋ	KoreanParticle	332.9959449

〈표 15〉 하위 10개의 Attention 을 갖는 형태소

형태소	품사	Attention 합
다	Josa	-295.5007837
쓰레기	Noun	-302.4656007
?	Punctuation	-375.3926609
최악	Noun	-525.9152848
재미없다	Adjective	-546.104634
...	Punctuation	-702.9990884
없다	Adjective	-838.8802801
..	Punctuation	-929.0254972
아깝다	Adjective	-940.3619517
.	Punctuation	-1606.767147

제 4 장 결론

본 연구에서 한국어를 수치화 하는 방법으로 자모, 글자, 단어, 형태소 단위로 벡터로 표현하는 방식을 제안하였고 이것을 Rotten Tomato Movie Review 와 네이버 영화 리뷰 데이터에 대해서 Attention Mechanism 을 적용한 순환신경망 모델에 대해 성능을 비교해 보았다. 네이버 영화 리뷰 데이터에서 단어 단위로 문서를 표현한 경우 모든 순환신경망 모델에서 가장 낮은 성능을 보였다. 실험에 사용하는 리뷰 데이터를 분석한 결과 단어로 문서를 표현한 경우 25%의 검증 데이터가 미등록어로 나타나는 것을 확인하였고 이것은 이것은 형태론적 활용이 다양한 한국어의 특성에 기인한 것으로 판단된다. 순환 신경망 모델 중에서는 Bi-directional LSTM 에 Attention Mechanism 을 적용한 모델에서 형태소를 이용한 문서 표현 방식에서 86.2%의 정확도를 보여 가장 좋은 성능을 보였지만 자모 및 글자로 문서를 표현한 경우에도 비슷한 정도로 우수한 성능을 나타냈다. 또한 입력 데이터에 대한 Attention Vector 를 구하는 실험으로부터 각각의 문서 단위가 감정 분석을 수행하는데 기여한 정도를 얻을 수 있었고 이를 기반으로 어떠한 형태소가 감성적 판단을 하는데 주요한 역할을 하는지 정량적으로 얻을 수 있었다. 최근 연구에서는 본 연구에 사용한 네이버 영화 리뷰 데이터에서는 평점은 높게 주었지만 리뷰는 부정적인 글을 남겼다거나, 악의적으로 별점을 낮추기 위해 의미없는 글을 남기는 경우도 포함이 되어 있었다. 좀 더 의미있는 데이터만 이용하여 실험에 사용한다면 더 좋은 성능을 보일 것으로 보인다. 본 연구에서는 문서를 긍정, 부정으로만 구분하였지만 실제 환경에서는 화남, 슬픔, 기쁨과 같은 다양한 감성적 표현이 존재할 수 있다. 더욱 다양한 감정 표현에 대한 감정 분석을 수행할 때 본 연구에서 사용한 Attention Mechanism 을 적용한 순환신경망 모델을 적용할 수 있을것이다.

참고문헌

- [1] Hinton, Geoffrey. "A practical guide to training restricted Boltzmann machines." Momentum 9.1, pp. 926, 2010.
- [2] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61, pp. 85–117, 2015.
- [3] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882, 2014.
- [4] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2, 2010.
- [5] Werbos, Paul J. "Backpropagation through time: what it does and how to do it." Proceedings of the IEEE 78.10, pp. 1550–1560, 1990.
- [6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8, pp. 1735–1780, 1997.
- [7] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555, 2014.

[8] Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems, 2017.

[9] Graves, Alex, Santiago Fernández, and Jürgen Schmidhuber. "Bidirectional LSTM networks for improved phoneme classification and recognition." Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, pp. 753–753, 2005.

[10] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473, 2014.

[11] Yang, Zichao, et al. "Hierarchical Attention Networks for Document Classification." HLT–NAACL, 2016.

[12] Wang, Yequan, et al. "Attention–based LSTM for Aspect–level Sentiment Classification." EMNLP, 2016.

[13] Britz, Denny, et al. "Massive exploration of neural machine translation architectures." arXiv preprint arXiv:1703.03906, 2017.

[14] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." International Conference on Machine Learning, 2015.

- [15] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*, 2013.
- [16] Sammons, Mark, et al. "EDISON: Feature Extraction for NLP, Simplified." *LREC*, 2016.
- [17] Lee, Jason, Kyunghyun Cho, and Thomas Hofmann. "Fully character-level neural machine translation without explicit segmentation." *arXiv preprint arXiv:1610.03017*, 2016.
- [18] Luong, Minh-Thang, Preslav Nakov, and Min-Yen Kan. "A hybrid morpheme-word representation for machine translation of morphologically rich languages." *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010.
- [19] Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 pp. 613–620, 1975.
- [20] Jean, Sébastien, et al. "On using very large target vocabulary for neural machine translation." *arXiv preprint arXiv:1412.2007*, 2014.

- [21] Olson, David L., and Dursun Delen. "Advanced data mining techniques" Springer Science & Business Media, pp. 138, 2008.
- [22] Pang, Bo, and Lillian Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005.
- [23] Hand, David J., and Keming Yu. "Idiot's Bayes—not so stupid after all?." International statistical review 69.3, pp. 385–398, 2001
- [24] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850, 2013.
- [25] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025, 2015.
- [26] 신수정, “감성 분석의 이해”, IT WORLD CIO IDG Tech Report, 2014.
- [27] twitter-korean-text, <https://github.com/twitter/twitter-korean-text>
- [28] Naver sentiment movie corpus v1.0, <https://github.com/e9t/nsmc>

[29] 네티즌 평점 : 네이버 영화 평점.리뷰 개봉 후 평점,
<http://movie.naver.com/movie/point/af/list.nhn>

[30] NLTK(Natural Language ToolKit), <http://www.nltk.org/>

Abstract

Sentiment Analysis and Its Applications with Attention-based Recurrent Neural Network

Hyuntak Joo

Department of Engineering Practice

Graduate School of Engineering Practice

Seoul National University

Natural Language Processing is the process of analyzing human language phenomena, transforming them into what the computer can understand and utilizing them. Especially Sentiment Analysis, a kind Natural Language Processing, aims to determine the attitude of subject with respect to some topic. The attitude includes a judgment or evaluation, or the emotional effect intended by the author. Also since Hangul has a different way of constructing letters than English, It needs to research how to convert Hangul to numerically analyzable form. This study deals with the method of effectively classifying the positive / negative evaluation of the Korean text based on Naver movie review data and compare with Rotten Tomato Movie Review, the English review dataset. The Recurrent Neural Network or its variations like Bi-directional Long-Short Term Memory or Attention Mechanism are used to training or validating the performance of the method of Hangul conversion. In experiments, Morpheme-level Bi-directional Long-Short Term Memory

with Attention Mechanism model shows the best score of 86.2% but Jamo-level model and word-level also shows good enough scores. Also, we can get morphemes affecting emotional judgment quantitatively by observing attention vector for the input document.

Keywords : Deep Learning, Natural Language Processing

Student Number : 2016-22258