



## 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](#)

석사학위논문

RNN을 이용한 한국어 감성분석:  
온라인 영화 후기를 중심으로

Sentiment Analysis based on Korean using  
Recurrent Neural Network: focused on Online Movie Review



국민대학교 일반대학원

데이터사이언스학과 데이터사이언스전공

이 재 준

2017

# RNN을 이용한 한국어 감성분석: 온라인 영화 후기를 중심으로

Sentiment Analysis based on Korean using  
Recurrent Neural Network: focused on Online Movie Review

지도교수 안 성 만

이 논문을 데이터사이언스학석사학위  
청구논문으로 제출함

2018 년 6 월 5 일

국 민 대 학 교 일 반 대 학 원

데이터사이언스학과 데이터사이언스전공

이 재 준

2017

이재준의

데이터사이언스학석사학위 청구논문을

인준함

2018년 7월 3일



심사위원장 권 순 범 인

심사위원 안 성 만 인

심사위원 정 여 진 인

국민대학교 일반대학원

# 차 례

그림 차례 .....	iii
표 차례 .....	iv
국문 요약 .....	v
제1장 서론 .....	1
제2장 관련연구 .....	4
2.1 감성분석 연구 .....	4
2.2 신경망을 이용한 감성분석 .....	6
2.3 Word2vec .....	10
2.4 RNN(Recurrent Neural Network) .....	12
2.4.1 Vanilla Recurrent Neural Network .....	12
2.4.2 LSTM(Long short-term memory) .....	14
2.4.3 GRU(Gated recurrent unit) .....	16
제3장 RNN을 이용한 한국어 감성분석 .....	19
3.1 음소 단위 감성분석 모형 .....	19
3.1.1 음소 단위 감성분석 모형의 전처리 .....	20
3.1.2 학습데이터를 이용한 음소 단위 감성분석 모형학습 .....	21
3.2 형태소 단위 감성분석 모형 .....	22
3.3 어절 단위 감성분석 모형 .....	24
제4장 실험 방법 .....	26
4.1 데이터 수집 및 전처리 .....	26

4.2 실험 종류 .....	30
<b>제5장 실험 결과 .....</b>	<b>33</b>
5.1 형태소 및 어절 단위 모형의 Word2vec 결과 .....	33
5.2 정확도 및 loss .....	34
5.2.1 음소 단위 모형의 정확도 및 loss .....	34
5.2.2 형태소 단위 모형의 정확도 및 loss .....	36
5.2.3 어절 단위 모형의 정확도 및 loss .....	39
5.3 학습시간 .....	41
<b>제6장 결론 및 향후 연구계획 .....</b>	<b>43</b>
<b>참 고 문 헌 .....</b>	<b>46</b>
<b>Abstract .....</b>	<b>53</b>



## 그림 차례

<그림 1> Recursive Neural Network .....	6
<그림 2> Bidirectional RNN .....	7
<그림 3> Model architecture .....	8
<그림 4> Word2vec의 CBOW와 Skip-gram 모형 .....	11
<그림 5> Vanilla Recurrent Neural Network .....	13
<그림 6> Long Short-Term Memory .....	15
<그림 7> Gated Recurrent Unit .....	17
<그림 8> 음소 단위 감성분석 프로세스 .....	19
<그림 9> 음소 단위 감성분석 모형의 진처리 과정 .....	20
<그림 10> 음소 단위 감성분석 모형의 학습과정 .....	21
<그림 11> 형태소 단위 감성분석 프로세스 .....	22
<그림 12> 어절 단위 감성분석 프로세스 .....	24
<그림 13> 입력 Padding 과정 .....	27
<그림 14> 형태소 출현 빈도수 및 형태소사전 .....	28
<그림 15> 어절 출현 빈도수 및 어절사전 .....	29
<그림 16> 음소 단위 모형의 te_loss 및 te_acc 변화 .....	35
<그림 17> 형태소 단위 모형의 te_loss 및 te_acc 변화 .....	38
<그림 18> 어절 단위 모형의 te_loss 및 te_acc 변화 .....	40
<그림 19> 모형 학습시간 .....	41

## 표 차례

<표 1> 비지도 및 지도학습 기반 감성분석 .....	5
<표 2> 한글 영화 후기를 이용한 감성분석 .....	9
<표 3> 신경망을 이용한 감성분석 요약 .....	9
<표 4> 수집한 영화 후기의 예 .....	26
<표 5> 음소사전 .....	27
<표 6> 데이터 요약 .....	30
<표 7> 실험 모형 .....	30
<표 8> 형태소 벡터의 분산표현 예시 .....	33
<표 9> 어절 벡터의 분산표현 예시 .....	33
<표 10> 음소 단위 모형 결과 .....	34
<표 11> 형태소 단위 모형 결과 .....	37
<표 12> 어절 단위 모형 결과 .....	39
<표 13> 모형 학습시간 .....	41



## 국문 요약

# RNN을 이용한 한국어 감성분석: 온라인 영화 후기를 중심으로

이 재 준

국민대학교 데이터사이언스학과

인터넷 발전으로 다른 사람들의 의견과 경험이 포함된 온라인 후기를 손쉽게 접할 수 있게 되었다. 후기는 제품과 서비스의 실질적 구매에 영향을 미치기 때문에 기업의 중요한 마케팅 전략으로 활용될 수 있다.

이에 언어에 포함된 의견과 태도와 같은 주관적인 정보를 탐지하는 감성분석 연구가 이루어졌다. 초기에는 감성 강도를 나타낸 감성사전을 통해 이루어졌으나 최근에는 딥러닝 기반 감성분석 연구가 증가하고 있다.

감성분석과 함께 말뭉치를 학습해 다음에 나올 단어 혹은 문장을 예측하는 언어모델에서도 딥러닝을 이용하고 있다. 한국어 문장을 이용한 언어모델 실험에서는 다양한 말뭉치를 활용하였지만, 감성분석은 형태소 단위 말뭉치만이 수행되었다.

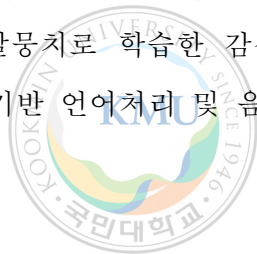
본 논문에서는 다양한 한국어 말뭉치를 이용해 문장의 감성을 긍정 또는 부정으로 판별하는 감성분석을 시도한다. 기존 감성분석에 사용한 감성사전을 구축하지 않고 문장을 구성하는 말뭉치로만 감성을 예측한다.

선행연구로 형태소 단위 모형과 함께 한글의 가장 작은 단위인 음소와 어절 단위 모형을 구축해 각 입력단위 별 모형성능을 비교분석 한다.

음소 단위 모형은 LSTM, Vanilla RNN, GRU와 같은 RNN layer를 3개 포함하는 모형을 구성하였다. 형태소 및 어절 단위 모형은 Word2vec을 이용해 말뭉치를 분산표현 한 후 LSTM layer를 2개 또는 3개 포함하는 모형을 구축하였다. 성능평가로는 정확도 및 loss를 측정하였고 각 모형별 학습시간을 비교하였다.

실험결과로는 음소 단위 모형은 LSTM 성능이 가장 뛰어났으며 형태소 및 어절 단위 모형에서는 LSTM layer가 모형성능에 영향을 미치지 않은 것을 확인하였다. 또한, 학습시간은 가장 작은 단위로 분할하여 문장길이 가 길어진 음소 단위 모형이 가장 크고 어절 단위 모형이 가장 작았다.

한국어 문장을 다양한 말뭉치로 학습한 감성분석 실험을 통해 영어에 비해 연구가 부족한 한글 기반 언어처리 및 음성인식 분야에 활용될 것으로 기대한다.



## 제1장 서론

다른 사람들이 생각하는 것은 의사결정과정에서 항상 중요한 부분으로 여겨졌다. 인터넷이 널리 퍼지기 전 사람들은 기계의 구매 평 또는 지역 선거 투표에 대한 견해를 듣기위해 주변 사람들에게 조언을 구했다. 하지만 인터넷의 발전으로 개인적 지인과 비평가가 아닌 사람들의 의견과 경험이 포함된 온라인 후기를 접할 수 있게 되었다(Pang and Lillian, 2008).

미국 성인 2000여 명을 대상으로 한 조사에 따르면, 인터넷 사용자의 81%는 적어도 한 번 제품의 온라인 후기를 검색했으며, 음식점 및 호텔과 같은 다양한 서비스의 후기를 접한 사람들의 87%는 후기가 구매에 큰 영향을 미쳤다고 응답했다(John, 2008). 또한, 국내 시장조사전문기관 마크로밀 엠브레인에 따르면 응답자의 73.8%는 광고에 비해 온라인 후기를 신뢰하며 상품 구매 전 다른 소비자의 후기를 검색한 응답자가 79%에 이른다(마크로밀 엠브레인, 2014). 이러한 후기는 소비자의 주관적 선호를 포함하며 실질적인 구매에 도움을 주기 때문에 기업의 마케팅 전략으로 사용할 필요가 있다고 한다(Zhu and Zhang, 2010).

이와 언어에 포함된 의견과 태도와 같은 주관적인 정보를 탐지하고 추출하는 방법인 감성분석에 대한 연구가 이루어졌다. 감성분석 연구는 2000년대 이후 급증하였는데 이는 웹사이트의 발전으로 인한 방대한 데이터의 이용가능성, 자연어 처리 및 정보 검색에 사용한 기계 학습의 증가에 있다(Pang and Lillian, 2008). 초기 감성분석은 긍정 및 부정을 내포하는 단어를 구축한 사전을 이용해 감성을 판별하였다(Turney, 2002;

Taboada et al., 2011). 대표적으로 영어 기반 감성사전인 SentiWordNet(SWN)이 존재하는데, 전체 어휘에 대한 긍정 및 부정 감성 강도를 이용해 다양한 분야에서 활용하였다(Feldman, 2013). 또한, 다양한 기계학습의 발전으로 이를 이용해 감성 분류기를 구축한 연구가 이루어졌으며(Pang et al., 2002) 최근에는 딥러닝(Deep Learning) 기반 감성분석의 연구가 증가하는 추세이다(Hu et al., 2015).

감성분석과 함께 말뭉치(corpus)를 학습해 주어진 문장에서 다음에 나올 적절한 글자, 단어 혹은 문장을 예측하는 언어모델(language model)에서도 딥러닝을 활용하고 있다. 한글 문장생성 실험에서 문장을 형태소, 어절, 음절 단위로 분할하여 입력 값으로 사용하였다. 실험 결과로는 음절 보다는 형태소 단위가, 형태소 단위 보다 어절 단위 모형의 성능이 뛰어났다(김양훈 et al., 2016). 하지만 이와 같이 말뭉치 사전의 크기가 큰 경우, 모델 복잡성이 커지며 사전에 포함되지 않은 어휘를 생성할 수 없는 단점이 존재한다(Lankinen et al., 2016). 이에 입력 값을 더 작은 단위인 음소로 분할하여 문장을 생성한 결과, 완성도가 높은 문장을 생성하였다(안성만 et al., 2017).

언어모델과 달리, 감성분석 연구는 주로 모형 입력을 주로 형태소 단위로 분할한 말뭉치를 사용하였다(김유영 et al., 2016; 권수정, 2017). 긍정과 부정을 나타내는 동사, 형용사, 부사를 분류하여 감성사전을 구축하였고 이를 이용해 감성분류기를 학습하였다(김유영 et al., 2016). 실험 결과로는 분류기 성능이 뛰어났지만, 사전구축 시 사용하는 오픈소스 형태소 분석기 성능에 따라 영향을 미쳤다. 딥러닝을 이용한 실험에서는 한글 문장을 형태소 분석기로 분해하여 말뭉치를 국소표현한 후 순환신경망을 이

용해 감성을 분류한 결과, LSTM을 이용한 모형 성능이 가장 뛰어났다 (권수정, 2017).

이에 본 논문에서는 한국어 영화 후기 문장을 각 음소, 형태소, 어절 단위로 분할한 후 말뭉치를 이용해 감성을 긍정 또는 부정으로 판별하는 감성분석을 시도한다. 이는 감성사전을 구축하지 않고 문장을 구성하는 말뭉치로만 학습을 진행하여 긍정 혹은 부정을 예측한다. 각 분할한 말뭉치를 분산표현 데이터로 가공한 후, 순환신경망을 이용해 감성분석 모형을 구축하고 한국어 문장을 다양한 방법으로 분할한 말뭉치 별 모형 특성을 살펴보고자 한다.

본 논문은 다음과 같이 구성하였다. 제 2장에서는 감성분석 연구, 신경망을 이용한 감성분석의 선행연구를 요약하고 모형 구축에 사용한 Word2vec, 순환신경망에 대해 서술한다. 제 3장에서는 음소, 형태소, 어절 단위 감성분석 모형을 설명한다. 제 4장 및 5장에서는 실험 방법과 실험 결과를 살펴보고, 마지막으로 제 6장은 본 연구에 대한 결론과 향후 연구 계획을 제시한다.

## 제2장 관련연구

### 2.1 감성분석 연구

감성분석은 다양한 기계학습을 이용하는데, 이때 기계학습은 크게 비지도학습(unsupervised learning)과 지도학습(supervised learning)에 기반을 둔 것으로 구분 할 수 있다(Pang and Lillian, 2010).

비지도학습 기반 감성분석은 긍정 또는 부정을 의미하는 단어를 미리 구축한 사전을 통해 감성을 판별하는 분석으로, 영어의 경우 대표적인 감성사전 SentWordNet(SWN)이 존재한다. 이는 전체 어휘에 대한 긍정 또는 부정 강도를 저장한 것으로 감성분석의 핵심 중 하나이다. 대부분 형용사 위주로 감성 강도를 측정하거나 문장 구조 및 어휘 품사 등을 고려한 자연어처리 기법, TF-IDF, 통계기법 등을 이용해 구축하였다(Scaffidi et al., 2007; Jin et al., 2009). 대표적인 비지도학습 기반 감성분석으로 Turney(2002)는 다양한 분야의 후기를 이용해 감성분석을 시도하였다. POS tagging을 이용해 문장안의 연속한 두 단어 쌍을 추출하는데, 하나는 후기의 성향을 나타내는 형용사나 부사이며 다른 하나는 문맥을 포함한 단어이다. 추출한 단어 쌍을 두 단어의 의미론적 연관성의 강도를 측정할 때 사용하는 PMI-IR 알고리즘(Church and Hanks, 1990)을 이용하여 긍정 및 부정에 대한 Semantic Orientation(이하 SO)값을 구한다. 이후 SO값의 평균을 이용하여 후기의 감성을 판별하였다.

지도학습 기반 감성분석은 감성분석 관련 연구들 중 가장 활발히 일어

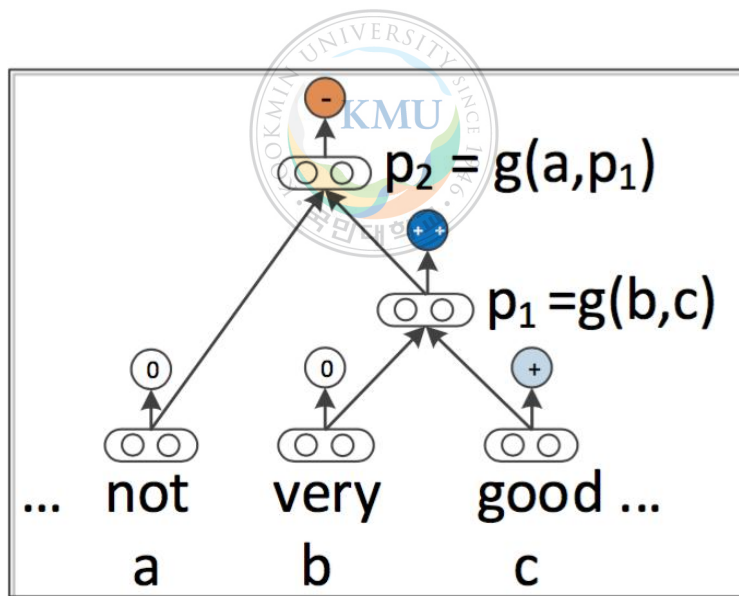
났으며(Appel et al., 2015), 지도학습 알고리즘을 이용해 감성분류기를 학습한다. 대표적으로 Pang et al.(2002)은 Naive Bayes, k-NN, Support Vector Machine(이하 SVM), Maximum Entropy를 이용해 감성 분류기를 학습하고 이의 성능을 비교하였다. 학습데이터의 표현은 unigrams, bigrams, POS tagging을 사용하고 각 분류기의 성능을 비교한 결과, Support Vector Machine이 가장 뛰어났다. Ferguson et al.(2009)은 금융 관련 블로그 데이터를 이용해 감성분석을 시도한 결과, Support Vector Machine 보다 Multinomial Naive Bayes 분류기의 성능이 더 뛰어남을 밝혔다. 또한, 긍정과 부정 이진분류 뿐 아니라 중립을 포함한 삼진분류 실험을 진행하였지만 이진 분류 보다 성능이 좋지 않았다. Pak과 Paroubek(2010)은 직접 사람이 감성을 분류한 트위터 데이터를 이용해 감성분석 시도하였다. Multinomial Naive Bayes로 학습 분류기를 구축하였고, 확률 언어 모델인 n-gram과 POS tagging을 이용한 데이터 표현에서 성능이 뛰어남을 확인했다. <표 1>은 비지도 및 지도학습 기반 감성분석 연구를 요약한 표이다.

<표 1> 비지도 및 지도학습 기반 감성분석

비지도학습 기반 감성분석	PMI 알고리즘과 POS tagging을 이용해 SO값의 평균값을 계산하여 감성을 판별(Turney, 2002)
지도학습 기반 감성분석	Naive Bayes, k-NN, SVM, Maximum Entropy를 이용해 분류기를 학습하고 이를 통해 감성을 판별(Pang et al., 2002)
	금융 관련 데이터와 Multinomial Naive Bayes을 이용한 감성분석(Ferguson et al., 2009)
	트위터 데이터와 Multinomial Naive Bayes을 이용해 감성을 판별하고 다양한 자질 표현으로 모형의 성능을 비교(Pak and Paroubek, 2010)

## 2.2 신경망을 이용한 감성분석

최근에는 머신러닝 기법 중 신경망이 발전한 형태인 딥러닝을 이용한 감성분석 연구가 이루어졌다. Socher et al.(2013)은 Recurrent neural network와 함께 자연어처리 분야에서 각광받는 Recursive neural network을 기반으로 한 Recursive Neural Tensor Network 모형을 구축하였다. Recursive neural network는 <그림 1>와 같이 입력 값으로 주어지는 몇 개의 단어를 묶어서 분석하는 트리구조를 지닌다(Goller and Kuchler, 1996).



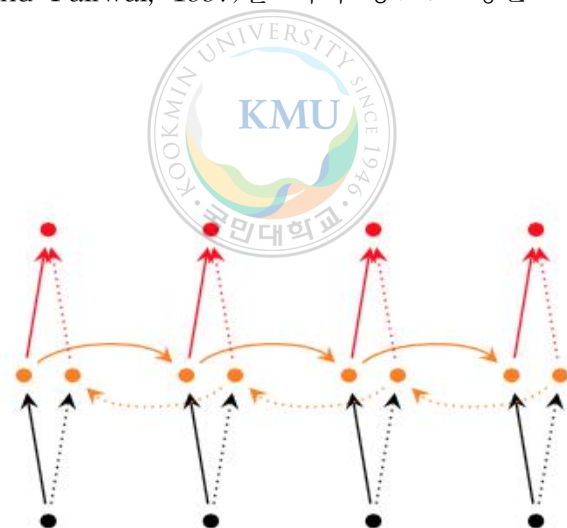
<그림 1> Recursive Neural Network(Socher, 2013)

<그림 1>에서  $b$ 와  $c$ 은 자식노드(child node)에 해당하는 단어벡터로 1차적으로 부모노드(parent node)  $p_1$ 으로 합쳐지고, 이후  $a$ 와 함께  $p_2$ 을 만



들어낸다.  $g$  은 두 단어벡터 간 연산을 의미하는 기호로 스코어와 벡터를 반환하는데, 감성분석의 경우 스코어는 문장의 긍정 또는 부정을 예측하는데 이용한다. 하지만 학습해야할 가중치와 편향의 수가 많으며 단어사전의 크기에 의존하는 문제점이 있다. 이에 모든 노드에 대해 동일한 텐서 기반 합성 함수를 사용한 Recurrent neural tensor network을 구축하여 감성분석을 시도하였다.

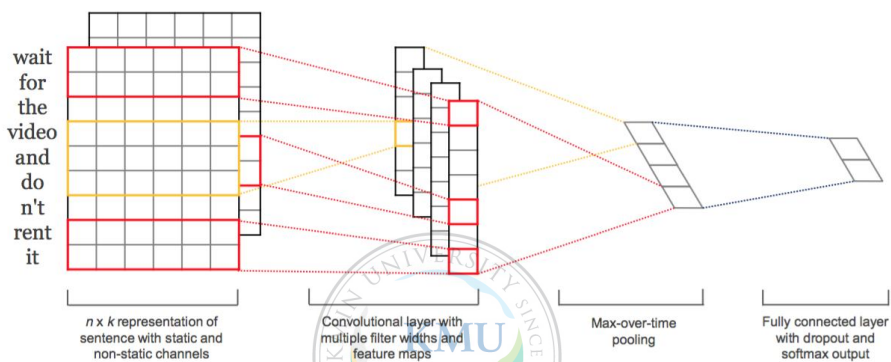
Irsoy와 Cardie(2014)는 감성분석의 한 종류로 문장에 포함된 의견을 탐지하는 실험에 순환신경망을 이용하였다. <그림 2>와 같이 양 방향으로 이전 시점 정보가 현 시점 정보에 영향을 주는 구조를 갖는 bidirectional RNN(Schuster and Paliwal, 1997)을 여러 층으로 쌓은 모델을 구축하였다.



<그림 2> Bidirectional RNN(Irsoy and Cardie, 2014)

Recurrent neural network 와 Bidirectional RNN을 다층으로 구축한 모형 성능을 비교한 결과, Bidirectional RNN으로 구축한 모형 성능이 더 뛰어났다.

이미지 인식 및 분류에 뛰어난 성능을 보이는 Convolutional Neural Network(이하 CNN)도 감성분석에 이용하였다. CNN은 여러 개 층에서 일반적 행렬 곱셈 대신 convolution을 사용하는 신경망이다(Goodfellow et al., 2016). 위치에 상관없이 시각적 특징을 추출하기 위해 제안되었으며 지역적 특징을 적용한 필터를 사용한다(LeCun et al., 1998).



<그림 3> Model architecture(Kim, 2014)

Kim(2014)은 <그림 3>와 같이 Word2vec을 이용한 단어벡터 채널(static)과 이를 미세하게 조정하는 채널(non-static) 2개를 입력 값으로 사용해 convolutional layer, max-over-time pooling, fully connected layer를 거쳐 영화 후기의 긍정 또는 부정을 예측하는 모형을 구축하였다. 이외 단어를 수치화 하는 random, static, non-static 등 3가지 방법을 적용해 모형의 성능을 비교하였으며, 영화 후기 이외 질문유형 분류 등 다양한 데이터를 이용한 문장 분류 실험도 함께 진행하였다. 영화 후기 데이터를 이용한 실험결과는 non-static만을 이용한 모형 성능이 뛰어났다.

위 연구들은 영어 문장을 사용한 반면, 권수정(2017)은 한글 영화 후기

를 사용해 Word2vec 및 순환신경망을 이용한 감성분석을 실험하였다. 한글을 형태소 단위로 분해하여 두 가지 표현방식을 이용해 <표 2>와 같이 5개 모형을 구축하였다.

<표 2> 한글 영화 후기를 이용한 감성분석(권수정, 2017)

no	표현방식	알고리즘
1	one-hot encoding	RNN
2	one-hot encoding	LSTM
3	one-hot encoding	GRU
4	one-hot encoding	k-NN
5	Word2vec	k-NN

실험데이터는 긍정 1만 개, 부정 1만 개로 총 2만 개 후기 데이터를 사용하였고 K-fold cross validation 방법을 사용해 정확도를 측정하였다. 실험결과로는 Word2vec과 k-NN을 이용한 감성분석 모형 정확도가 가장 뛰어났다. 순환신경망 중 LSTM을 사용한 모형 정확도가 가장 높았지만 GRU와 큰 차이를 보이지 않았다. 다음 <표 3>는 신경망을 이용한 감성분석 선행연구를 간략히 표로 정리한 것이다.

<표 3> 신경망을 이용한 감성분석 요약

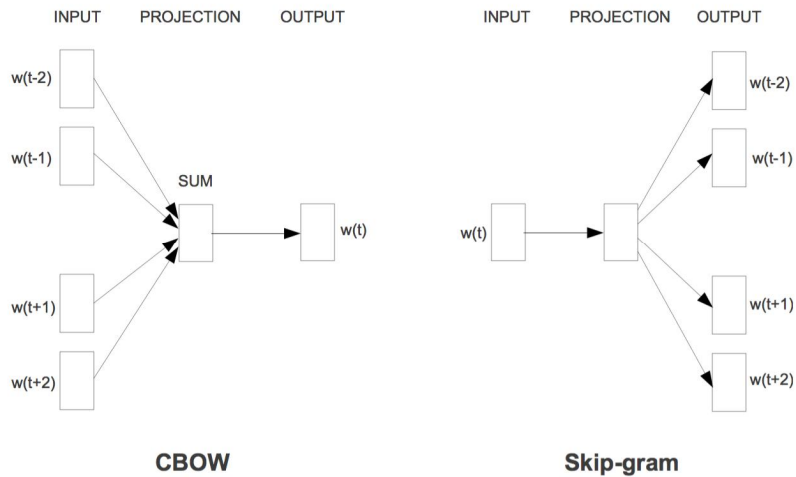
신경망을 이용한 감성분석	Recursive neural network을 이용한 감성분석 (Socher et al., 2013)
	CNN을 이용한 감성분석(Kim, 2014)
	Bidirection-RNN을 이용한 문장 속 의견 탐지 (Irsoy and Cardie, 2014)
	Word2vec과 순환신경망을 이용한 감성분석(권수정, 2017)

한국어 문장을 이용한 감성분석에서는 말뭉치를 형태소 단위로 사용하였다. 이에 본 논문에서는 감성사전을 구축하지 않고 한글 문장을 음소, 형태소, 어절 단위로 분해한 후 2.4절에서 소개하는 RNN을 이용해 감성을 예측하는 모형을 구축하고자 한다.

### 2.3 Word2vec

텍스트를 수치로 변환하는 방법은 국소 표현(one-hot representation)과 분산 표현(distributed representation) 방법이 있다. 국소 표현은 해당 단어가 존재하는 위치에만 1로 표현하며, 분산 표현은 단어를 표현하는 수치 값이 모두 존재하지만 서로 다른 단어는 값을 다르게 설정한다.

Word2vec은 분산 표현 방법 중 하나로 말뭉치를 입력받아 각 단어를 벡터로 표현하는데, 이와 같이 단어의 의미와 맥락을 고려하여 벡터로 표현하는 것을 word embedding이라 한다. Mikolov et al.(2013)은 <그림 4>와 같이 word embedding 방법 중 하나인 Word2vec의 두 학습모형을 제안하였다. input layer, projection layer, output layer 3개 층으로 구성된 모형으로 순환신경망을 도입하였다. CBOW는 주변 단어들을 이용한 단어를 예측하며, 이와 반대로 Skip-gram은 한 단어를 이용해 주변 단어를 예측하는 구조를 지닌다. 모형 입력 값은 국소 표현 방식을 사용하며 모형 결과 값과 실제 단어 벡터의 오류를 최소화하며 학습을 진행한다.



<그림 4> Word2vec의 CBOW와 Skip-gram 모형(Mikolov et al., 2013)

Word2vec은 학습을 통해 연관된 단어들은 근접할 가능성이 높으므로 가까운 벡터 값을 가질 확률이 증가한다. 이렇게 학습한 단어는 연관된 단어들끼리 비슷한 벡터 값을 갖으며, 단어 벡터의 차원을 축소할 수 있다. Word2vec을 이용한 한글 개체명 인식 실험에서는 단어 벡터의 차원을 다양하게 축소한 결과, 축소한 차원수가 높을수록 성능이 뛰어난을 밝혔다(한남기, 2016). 이에 본 논문에서는 형태소 및 음절 단위 모형 구축 시 Word2vec의 Skip-gram을 이용해 벡터의 차원을 99로 축소하고자 한다.

## 2.4 RNN(Recurrent Neural Network)

### 2.4.1 Vanilla Recurrent Neural Network

RNN(Recurrent Neural Network)은 문자와 음성과 같은 연속된 정보를 처리하는데 적합한 신경망 모형이다(Goodfellow et al., 2016). 기존 인공 신경망(Artificial Neural Network)의 각 입력 값들은 독립적으로 사용한다. 그러나 문장 또는 음성의 경우  $t$ 시점의 입력 값은  $t+1$ 시점 입력 값에 영향을 준다. 이러한 특징을 고려하여 순환신경망의 은닉계층(hidden layer)은 각 시점의 입력 값이 서로 영향을 주는 재귀적인 구조로 이루어졌다. 은닉계층 및 출력계층 노드 값은 아래와 같은 방법으로 계산한다.

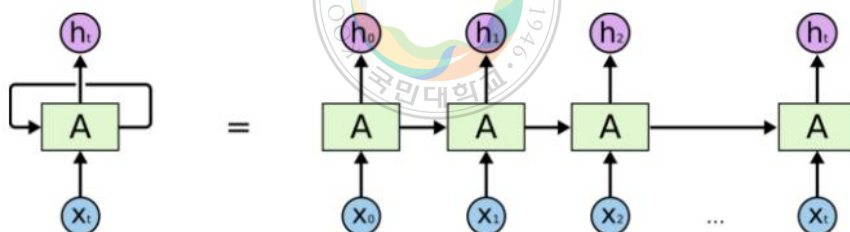
$$h_t = f(Ux_t + Wh_{t-1} + b)$$

$$o_t = g(Vh_t + c)$$

은닉계층의 노드 값( $h_t$ )은  $t$ 시점의 입력 값( $x_t$ )과 이전 시점의 은닉계층 노드 값( $h_{t-1}$ )의 선형결합을 활성화 함수( $f$ )를 이용하여 표현한다. 활성화 함수는 sigmoid, tanh, ReLU(Rectified Linear Unit)와 같이 비선형 함수를 사용한다. 이후 출력계층의 노드 값( $O_t$ )은 softmax 또는 sigmoid와 같은 활성화 함수( $g$ )를 이용하여 0과 1사이의 확률 값으로 변환시켜 예측한다. 이는 문장 생성 실험의 경우 다음 출현할 단어를 예측하고, 감성분석

의 경우 긍정 또는 부정을 예측한다. 식에서 사용한  $U, W, V$  은 가중치를 나타내는 행렬이며  $b, c$  은 선형결합에 사용한 편향을 의미한다.

모형으로부터의 예측 값이 실제 데이터와 얼마나 정확한지 평가하기 위해 cross entropy 비용함수를 목적함수(Cost function)로 설정한다. 이후 목적함수를 최소화시키는 가중치와 편향을 최적화 알고리즘을 통해 찾는다. 이 중 gradient descent(Cauchy, 1847)는 목적함수를 gradient를 이용해 최솟값을 찾아가는 방법이지만, 모든 데이터를 사용하기 때문에 연산량이 방대해져 속도가 느려진다. 이를 개선하기 위해 일부 데이터를 이용하여 가중치와 편향을 찾아내는 stochastic gradient descent 알고리즘을 주로 사용한다. 더불어 출력 층에서부터 gradient를 계산하여 가중치와 편향을 학습하는 역전파 알고리즘(back-propagation)을 이용한다.



<그림 5> Vanilla Recurrent Neural Network(Olah, 2015)

<그림 5>은 Vanilla RNN cell을 확장시킨 모습을 표현한 것이다. 모형의 출력 값은 무한한 과거 시점의 출력에 의존(long-term dependency)하는 것을 확인할 수 있다. 이는 역전파 알고리즘을 이용하여 파라미터를 학습할 때, 목적함수의 편미분 계수가 0에 가까워지거나(vanishing gradient) 매우 커지는 문제가 발생한다(Bengio et al., 1993). 편미분 계수

가 0에 가까워지면 파라미터 학습이 제대로 이루어지지 않으며, 매우 커지게 되면 학습이 불안정해진다.

#### 2.4.2 LSTM(Long short-term memory)

RNN 모형 중 하나인 LSTM(Long short-term memory)은 vanilla RNN과 비슷한 형태를 가지지만 은닉계층이 memory cell이라 불리는 구조를 지닌다(Hochreiter and Schmidhuber, 1997). 이는 오래된 시점의 출력 값과 현 시점의 입력 값을 어느 정도 반영할지 게이트(gate)를 이용해 학습하도록 고안되었다. 세 가지 게이트(gate)를 가지며 다음과 같이 Vanilla RNN에 비해 상대적으로 복잡한 식을 가진다.

$$\bar{c}_t = \tanh(x_t U^c + h_{t-1} W^c)$$

$$i_t = \sigma(x_t U^i + h_{t-1} W^i)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f)$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o)$$

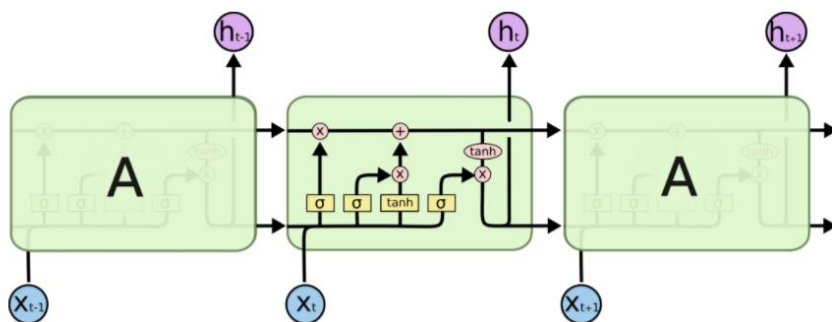
$$c_t = \bar{c}_t \circ i_t + c_{t-1} \circ f_t$$

$$h_t = \tanh(c_t) \circ o_t$$

현 시점의 입력 값( $x_t$ )과 이전 시점 은닉노드 출력 값( $h_{t-1}$ )의 선형결합



을 활성화 함수( $\sigma$ )를 통해 내부 은닉노드 후보 값( $\bar{c}_t$ )을 계산한다. 이후 입력 게이트(input gate), 망각 게이트(forget gate), 출력 게이트(output gate) 출력 값  $i_t, f_t, o_t$  은 이전 시점 은닉노드 출력 값( $h_{t-1}$ )과 현 시점 입력 값( $x_t$ )의 선형결합을 활성화 함수( $\sigma$ )를 거쳐 0과 1사이 값으로 생성한다. 입력 게이트 출력 값  $i_t$ 은 내부 은닉노드 후보 값( $\bar{c}_t$ )과 결합하여 이를 얼마큼 통과시킬지 결정한다. 망각 게이트 출력 값  $f_t$ 은 이전 시점 내부 은닉노드 값( $c_{t-1}$ )과 결합하여 이를 얼마큼 반영할지 결정한 후  $\bar{c}_t \circ i_t$  와 선형결합 하여 현 시점 내부 은닉노드 값( $c_t$ )을 생성한다. 현 시점 내부 은닉노드 값( $c_t$ )은  $\tanh$ 를 통과한 후 출력 게이트 출력 값( $o_t$ )과 결합하여 다음 네트워크로 얼마큼 반영할지 계산하여 최종 은닉노드 출력 값( $h_t$ )을 생성한다. 식에서 사용한  $U, W$  은 가중치를 나타내는 행렬이며 위첨자  $c, i, f, o$  은 은닉노드 후보 값( $\bar{c}_t$ ), 입력게이트, 망각 게이트, 출력 게이트에서의 계산을 의미한다.  $\circ$  은 벡터 간 각 요소의 곱이며  $\sigma$  은 활성화 함수로 LSTM에서는 주로 sigmoid 을 사용한다.



<그림 6> Long Short-Term Memory(Olah, 2015)

<그림 6>은 LSTM의  $t$ 시점의 memory cell을 모형화한 것이다.

Vanilla RNN에 비해 복잡한 계산과정으로 학습이 필요한 파라미터가 증가하므로, 학습시간은 증가하지만 각 gate로부터 오래된 시점의 출력 값에 선택적으로 의존한다. 이는 Vanilla RNN에서 발생하는 무한한 과거 시점 출력 값에 의존함을 줄일 수 있다.

### 2.4.3 GRU(Gated recurrent unit)

LSTM의 변형모형 중 하나인 GRU(Gated recurrent unit)는 LSTM과 유사한 구조를 지니지만 gate에서 차이를 보인다(Cho et al., 2014). 리셋 게이트(reset gate)와 업데이트 게이트(update gate)로 이루어져있으며 리



$$r_t = \sigma(x_t U^r + h_{t-1} W^r)$$

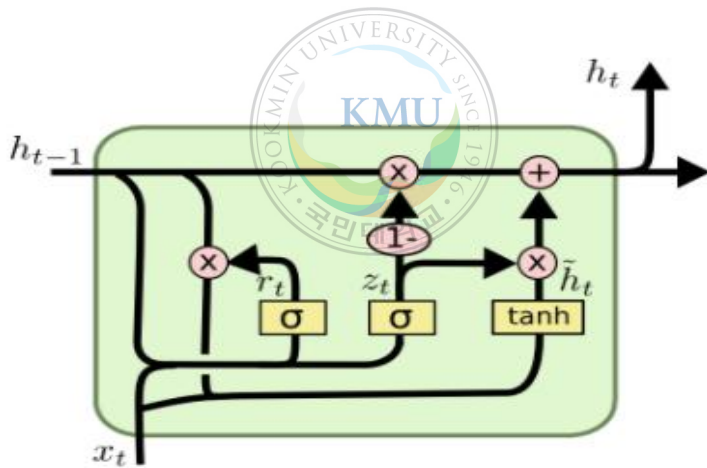
$$u_t = \sigma(x_t U^u + h_{t-1} W^u)$$

$$\bar{h}_t = \tanh(x_t U^h + (r_t \circ h_{t-1}) W^h)$$

$$h_t = \bar{h}_t \circ (1 - u_t) + h_{t-1} \circ u_t$$

현 시점 입력 값( $x_t$ )과 이전 시점 은닉노드 출력 값( $h_{t-1}$ )의 선형결합을 활성화 함수( $\sigma$ )를 통해 0과 1사이의 값을 갖도록 리셋 게이트, 업데이트

출력 값  $r_t, u_t$  을 계산한다. 리셋 게이트 출력 값( $r_t$ )과 이전 시점 은닉노드 출력 값( $h_{t-1}$ )의 곱합( $r_t \cdot h_{t-1}$ )을 현 시점 입력 값( $x_t$ )과 선형결합 하여  $\tanh$ 을 통해 내부 은닉노드 후보 값( $\bar{h}_t$ )을 계산한다. 이후 내부 은닉노드 후보 값( $\bar{h}_t$ )과  $1-u_t$ 의 곱합은 이전 시점 은닉노드 출력 값( $h_{t-1}$ )과 업데이트 게이트 출력 값( $u_t$ )의 곱합과 더해져 최종 은닉노드 출력 값( $h_t$ )을 생성한다. 식에서 사용한  $U, W$  은 가중치를 나타내는 행렬이며 위첨자  $r, u, h$  은 내부 은닉노드 후보 값( $\bar{h}_t$ ), 업데이트 게이트, 리셋 게이트에서의 계산을 의미한다.  $\cdot$  은 벡터 간 각 요소의 곱을 나타내며  $\sigma$  은 활성화 함수로 GRU 모형에서도 sigmoid 을 사용한다.



<그림 7> Gated Recurrent Unit(Olah, 2015)

<그림 7>은 GRU의  $t$  시점을 모형화한 것이다. GRU의 업데이트 게이트는 LSTM의 입력 게이트와 망각 게이트의 역할을 하며 최종 은닉노드 출력 값( $h_t$ )은 LSTM과 다르게 활성화 함수를 거치지 않는다. 또한 gate 수가 LSTM에 비해 적어 학습할 파라미터 수가 줄어들며 학습시간도 단

측된다.

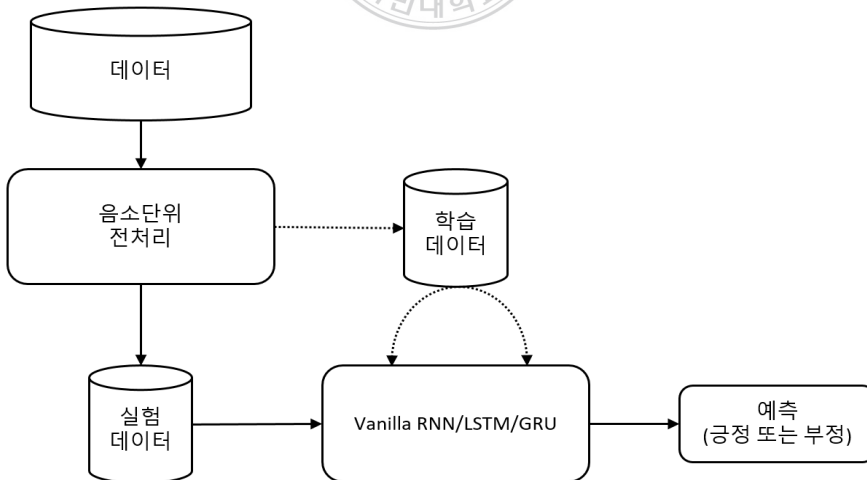
LSTM과 GRU의 성능차이는 Chung et al.(2014)과 Jozefowicz et al.(2015)에 의해 큰 차이가 없다고 한다. GRU의 경우 가중치와 편향의 수가 LSTM에 비해 적어 소규모 데이터로도 학습이 가능하지만, 데이터 규모가 커지는 경우 LSTM의 결과가 뛰어날 수 있다. 본 절에서 소개할 RNN을 이용한 감성분석 모형 구축은 3장에서 다루도록 한다.



## 제3장 RNN을 이용한 한국어 감성분석

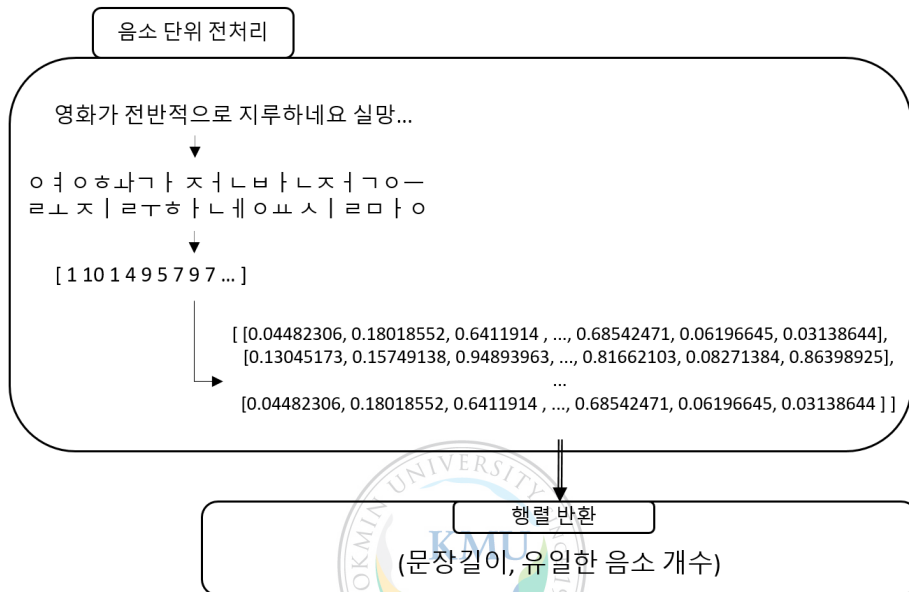
### 3.1 음소 단위 감성분석 모형

본 논문에서는 한국어 문장을 음소 단위로 분할한 뒤 입력하여 문장의 감성을 부정 또는 긍정으로 예측하는 모형을 구축한다. <그림 8>는 음소 단위 감성분석 모형의 전체과정을 간략히 그림으로 표현한 것이다. 수집한 영화 후기 문장들은 음소 단위 전처리 과정을 거쳐 수치화한다. 이후 학습데이터와 실험데이터로 분류하고 학습데이터를 이용해 음소 단위 감성분석 모형을 학습한다. 학습이 완료된 후 실험데이터를 입력해 문장의 감성을 잘 분류하는지 정확도를 비교하며 모형 성능을 평가한다.



<그림 8> 음소 단위 감성분석 프로세스

### 3.1.1 음소 단위 감성분석 모형의 전처리



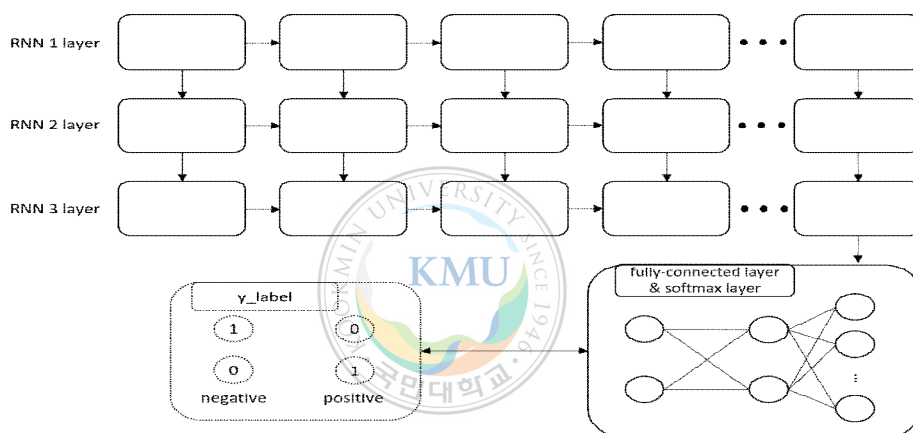
<그림 9> 음소 단위 감성분석 모형의 전처리 과정

본 논문에서 구축한 음소 단위 감성분석 모형은 전처리 단계를 거치며 <그림 9>와 같이 그림으로 표현하였다. 예를 들어 영화 후기 “영화가 전반적으로 지루하네요 실망..”은 “○,ㄷ,○,ㅎ,ㅏ,ㅓ,ㅑ,ㅓ, ㅈ,ㅣ,ㄴ,비,ㅓ,ㄴ,ㅈ, ㅣ,ㅑ,○,ㅡ,ㄹ,ㅏ, ㅈ,ㅣ,ㄹ,ㅓ,ㅎ,ㅓ,ㄴ,ㅋ,○,ㅠ, ㅅ,ㅣ,ㄹ,ㅓ,ㅓ,ㅇ, ..”와 같이 음소 단위로 분해한다. 이는 모든 문장을 음소 단위로 분해하여 구축한 사전을 통해 문장벡터로 변환한다.

형태소 또는 음절과 달리 한글의 가장 작은 단위인 음소는 의미를 갖지 않으므로 각 음소를 나타내는 벡터는 임의의 값으로 설정하였다. 따라

서 최종 모형 입력은 문장길이와 유일한 음소 개수로 이루어진 2차원 행렬로 표현한다. <그림 9>에서 문장길이는 한 후기의 포함된 음소 개수, 유일한 음소 개수는 구축한 음소사전의 크기를 나타낸다.

### 3.1.2 학습데이터를 이용한 음소 단위 감성분석 모형학습



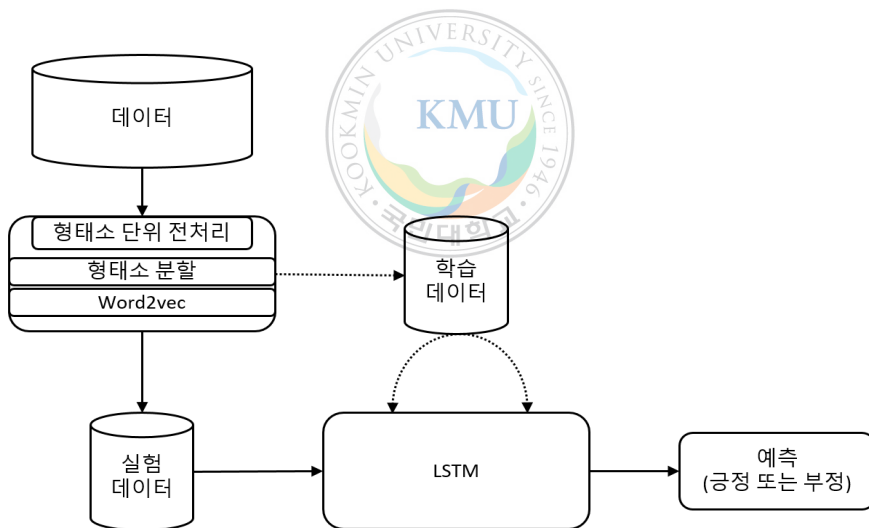
<그림 10> 음소 단위 감성분석 모형의 학습과정

음소 단위 감성분석모형의 학습과정을 <그림 10>과 같이 세부적으로 나타냈다. 후기 문장의 음소를 나타내는 벡터들은 첫 번째 RNN을 통과하는데, 순차적으로 입력하며 각 시점의 출력 값을 생성하고 이는 다음 RNN의 입력으로 사용한다. 이와 같은 방법으로 3개 RNN을 통과한 후 마지막 시점의 출력 값을 추출한다. 이는 같은 층의 뉴런들이 서로 연결하여 가중치와 편향으로 새로운 값을 전달하는 신경망의 일반적인 층인 fully-connected layer을 이용해 크기가 2인 벡터를 출력한다. 이후 softmax layer를 통해 문장의 감성이 부정과 긍정이 될 확률을 계산하고,

y\_label를 이용해 목적함수(cost function)를 최소화하며 학습을 진행한다.

이와 같은 학습을 반복적으로 진행한 후, 전처리 과정을 거쳐 분류한 실험데이터를 이용해 성능을 확인한다. 본 논문에서는 3장에서 소개한 Vanilla RNN, LSTM, GRU을 이용해 음소 단위 모델을 구축하고 비교분석한다.

### 3.2 형태소 단위 감성분석 모형



<그림 11> 형태소 단위 감성분석 프로세스

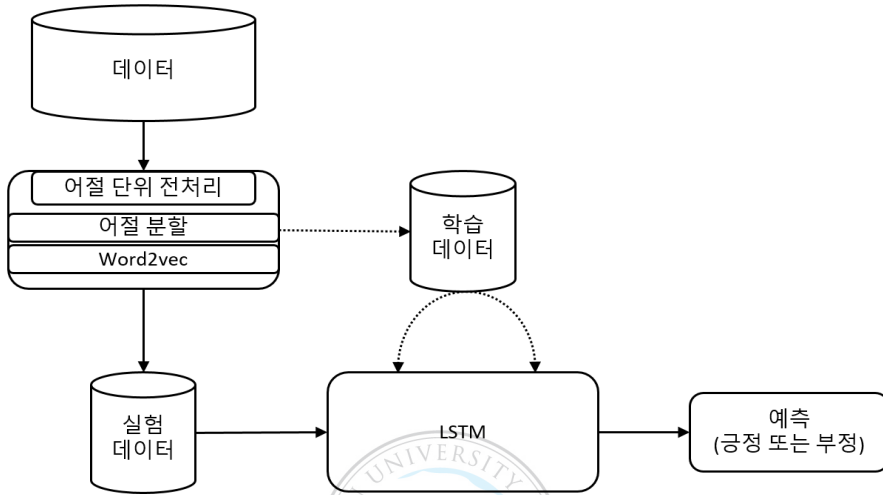
본 논문에서 형태소 단위 감성분석모형을 <그림 11>과 같이 구축한다. 예를 들어 영화 후기 “영화가 전반적으로 지루하네요 실망..”은 “영화, 가, 전반, 적, 으로, 지루, 하, 네요, 실망, ..”과 같이 형태소 단위로 분해한다.



이후 각 형태소마다 출현 빈도를 계산해 일정 횟수 미만으로 출현한 형태소를 제외한 사전을 구축하고, 이를 통해 문장을  $k$ 개의 형태소로 구성한 벡터  $x_n = (c_1, c_2, \dots, c_k)^T$ 로 변환한다.  $x_n$ 은  $n$ 번째 문장을 나타내는 문장 벡터,  $c_i$  ( $1 \leq i \leq k$ )은 사전을 이용하여 형태소를 표현한 자연수,  $T$ 은 열벡터를 의미한다.

이후 문장벡터는 2.3절에서 소개한 Word2vec의 Skip-gram 모델을 이용해 각 형태소의 고차원 벡터를 저차원 벡터로 변환하고 최종적으로 2차원 행렬  $(p, q)$ 로 나타낸다.  $p$ 은 후기에 포함된 형태소 개수,  $q$ 은 embedding 과정을 거쳐 축소된 차원을 수를 나타낸다. 형태소 단위 감성 분석 모델의 학습과정은 음소 단위 모델과 동일하다. 본 논문에서는 선행 연구에서 성능이 뛰어난 LSTM을 이용해 모델을 구성하고(권수정, 2017), LSTM 층을 여러 개 포함한 모델을 구축해 성능을 비교한다.

### 3.3 어절 단위 감성분석 모형



<그림 12> 어절 단위 감성분석 프로세스

본 논문에서는 어절 단위 감성분석모형을 <그림 12>과 같이 구축한다. 예를 들어 영화 후기 “영화가 전반적으로 지루하네요 실망..”은 “영화가, 전반적으로, 지루하네요, 실망, ..”과 같이 어절 단위로 분해한다. 이는 모든 문장을 어절 단위로 분해하여 구축한 사전을 통해 문장을  $k$ 개의 어절로 구성된 벡터  $x_m = (w_1, w_2, \dots, w_l)^T$ 로 변환한다.  $x_m$ 은  $m$ 번째 후기를 나타내는 문장벡터,  $w_j$  ( $1 \leq j \leq k$ )은 사전을 이용하여 어절을 표현한 자연수,  $T$ 은 열벡터를 의미한다.

이후 형태소 단위 모형과 같은 방법으로 Word2vec의 Skip-gram 모형을 이용해 최종 입력 값을 2차원 행렬  $(u, v)$ 로 표현한다.  $u$ 은 후기에 포함된 어절 수,  $v$ 은 embedding과정을 거쳐 축소된 차원을 나타낸다. 어

절 단위 감성분석의 학습과정은 음소 및 형태소 단위 감성분석과 동일하다. 본 논문에서는 선행 연구에서 성능이 뛰어난 LSTM을 이용해 모델을 구성하며(권수정, 2017), 형태소 단위 모형과 마찬가지로 LSTM 층을 여러 개 포함한 모델을 구축해 성능을 비교한다.



## 제4장 실험 방법

### 4.1 데이터 수집 및 전처리

실험에 사용할 데이터는 네이버 영화 후기 문장을 수집해 별의 개수가 1개 이상이고 5개 이하는 부정, 6개 이상이고 10개 이하는 긍정으로 분류하였다. <표 4>은 모형에 사용한 실험데이터를 정리한 표이다. 2015년부터 2017년까지 개봉한 영화들로 영화당 페이지를 20으로 제한하여 수집하였다. 총 20만 건 이상 수집하였지만, 대체로 부정보다 긍정 후기의 수가 많았다. 이에 긍정과 부정 후기 비율을 50%로 맞춰 실험에 사용할 데이터를 90,000건으로 설정하였다. 모형 학습에 사용할 학습데이터와 성능 확인을 위한 실험데이터의 비율은 7:3으로 설정하였다.

<표 4> 수집한 영화 후기의 예

	영화 후기
긍 정	<ul style="list-style-type: none"><li>- 재미있게 잘 봤습니다.^^</li><li>- 오랜만에 사춘기의 설레임을 느낄 수 있었습니다.극장판이라서 그런지 좀 짧아서 아쉽습니다.TV판이나 코믹스가 나오길 기대합니다.</li><li>- 굳굳굳굳 굳 여자친구랑 봤는데 꿀잼 나쁘지않음</li></ul>
부 정	<ul style="list-style-type: none"><li>- 1편보다 못함. 긴장감도 없고조별로</li><li>- 너무 힘들었다 한매 한매</li><li>- 이걸 영화가 아니라 연휴 금요일 밤 7시에 지상파에서 틀어도 시청률 0.001프로도 안나올듯</li></ul>

음소 단위 모형의 학습에 사용할 데이터를 문장벡터로 변환하기 위해 모든 후기를 음소 단위로 분해하여 51개 유일한 음소로 이루어진 사전을 구축하였다. <표 5>는 음소사전을 나타내며 이를 이용하여 각 문장에 사용한 음소  $i$  ( $1 \leq i \leq 51$ )을 수치로 변환한다. 또한, 문장벡터의 길이를 동일하게 설정하기 위해 수집한 문장 최대길이인 343으로 설정하였고 이보다 길이가 작은 후기는 <그림 13>와 같이 0으로 padding하였다.

<표 5> 음소사전

음소	값
ㅅ	1
ㅈ	2
ㅇ	3
ㄱ	4
...	...
라	51

ㅈ | ㄱㅈㅎ | ㄴ ㅈ | ㅇ ㅈ  
 [ 10, 25, 34, 27, 1, 6, 8, 11, 7, 48 ]

(문장길이 10)



[ 10, 25, 34, 27, 1, 6, 8, 11, 7, 48, 0, 0, 0, 0, ..., 0 ]

( 후기 문장 길이의 최댓값 343로 맞추기 위해 0 으로 Padding )

<그림 13> 입력 Padding 과정

형태소 단위 모형의 학습에 사용할 데이터를 문장벡터로 변환하기 위해

모든 후기를 형태소 단위로 분해한 결과, 42446개 유일한 형태소가 존재했다. 문장에 포함된 모든 형태소를 사용하게 되면, 출현 빈도가 1인 형태소가 포함된 후기는 성능에 큰 영향을 미친다. 따라서 형태소 마다 출현 빈도를 계산하고 빈도가 2미만 인 형태소를 제외한 사전을 <그림 14>와 같이 구축하였으며, 사전 크기는 10000으로 설정하였다.

( '는', 34704 )	형태소	값
( '영화', 29178 )	는	1
( '은', 16416 )	영화	2
( '재미있', 5544 )	은	3
( '지루', 5527 )	재미있	4
...	...	
( '버러지', 3 )	버러지	10000
...		

<그림 14> 형태소 출현 빈도수 및 형태소사전

이를 이용해 각 후기에 속한 형태소  $j$  ( $1 \leq j \leq 10000$ ) 을 자연수로 수치화하여 문장벡터로 변환한다. 또한, 문장벡터의 길이를 동일하게 설정하기 위해 음소 단위 모형에 사용한 입력과 같은 방법으로, 수집한 문장의 최대길이인 87로 설정하고 이보다 적은 후기는 0으로 padding하였다. 이후 Word2vec을 이용해 거쳐 10000 차원 형태소를 99로 축소하였고 모형 입력은 (87, 99) 행렬로 변환하였다.

어절 단위 모형의 학습에 사용할 데이터를 문장벡터로 변환하기 위해 모든 후기를 어절 단위로 분해한 결과, 125309개 유일한 어절이 존재하였



<표 6> 데이터 요약

	음소	형태소	음절
데이터 수	90000(긍정 : 45000, 부정 : 45000)		
수집 경로	네이버 영화 리뷰 <sup>1)</sup>		
학습-실험데이터 비율	7:3 (train : 63000, test : 27000)		
후기 길이의 최댓값	343	87	37
사전 크기	51	10000	40000

## 4.2 실험 종류



<표 7> 실험 모형

No	입력 단위	표현방식	순환신경망 모형	layer 수	최적화 알고리즘
1	음소	random vector	LSTM	3	Adam
2			Vanilla RNN	3	
3			GRU	3	
4	형태소	Word2vec	LSTM	1	
5				2	
6				3	
7	어절	Word2vec	LSTM	1	
8				2	
9				3	

1) <https://movie.naver.com/movie/point/af/list.nhn>



실험은 <표 7>와 같이 RNN, LSTM, GRU를 이용한 음소 단위 감성분석 모형과 Word2vec과 LSTM을 이용한 형태소 및 어절 단위 감성분석 모형을 구축하였다. 말뭉치 표현방식은 음소 단위 모형의 경우, 임의 값으로 설정하였고 형태소 및 어절 단위 모형은 Word2vec을 이용하였다. 실험에 사용한 최적화 알고리즘으로는 local minimum으로 수렴 가능한 stochastic gradient descent의 변형모형 중 하나로 Adam(Kingma and Ba, 2014)을 사용하였다. 학습 중 발생할 수 있는 과적합(overfitting)을 방지하기 위해 순환신경망 layer 각 층에서 50%를 dropout(Srivastava et al., 2014) 하였고, 실험데이터로 측정한 모형성능이 5 epoch 이내에 향상하지 않으면 학습을 중지하였다.

성능평가로는 정확도, loss, 학습시간 3가지를 비교하였다. 정확도는 전체 데이터 중 예측 값이 실제 값과 같은 True Positive(실제 긍정인데 긍정으로 예측한 수)와 True Negative(실제 부정인데 부정으로 예측한 수)의 비율로 아래와 같은 식을 이용해 계산한다.

$$Accuracy(\text{정확도}) = \frac{TP + TN}{TP + TN + FP + FN}$$

*FP*: 실제값이 부정인데, 긍정으로 예측한 수  
*FN*: 실제값이 긍정인데, 부정으로 예측한 수

loss는 모형의 목적함수(Cost function)로 설정한 cross entropy를 이용한 값으로 실제 값과 예측 값의 차이를 나타내며, 0에 가까울수록 학습이 잘 되었다 판단할 수 있다. 학습시간은 실험 데이터를 이용해 모형을 학습한 시간으로 초 단위로 측정하였다. 모든 실험은 Intel Xeon CPU 3개

(48코어)와 Nvidia GeForce GTX 1080Ti GPU 1개(16%)가 설치된 서버에서 진행하였다. 코드는 파이썬 딥러닝 프레임워크 중 하나인 Tensorflow(Abadi et al., 2016) 패키지를 활용하였다.



## 제5장 실험 결과

### 5.1 형태소 및 어절 단위 모형의 Word2vec 결과

형태소 및 어절 단위 감성분석 모형의 입력 값은 Word2vec을 이용해 분산표현으로 변환하였다. Skip-gram 모형을 사용하였고 학습 반복은 10000, 예측할 형태소는 바로 앞뒤로 예측하도록 window 사이즈를 1로 설정하였다. <표 8>와 <표 9>은 모든 후기 문장을 각각 형태소와 어절로 분할하여 Word2vec을 통해 생성한 벡터를 분산 표현한 것이다. 형태소 및 어절 차원을 99로 축소하였고 생성한 벡터를 감성분석 모형의 입력 값으로 사용하였다.

<표 8> 형태소 벡터의 분산표현 예시

	1	2	3	...	98	99
‘이’	0.4824731	0.5286429	-0.388828	...	-0.035862	-0.810023
‘는’	-0.215001	-0.335167	0.3011711	...	-0.544502	-0.568692
‘고’	-0.620236	-0.008427	0.3192004	...	-0.117471	-0.135857
‘영화’	-0.051262	-0.503489	0.4363528	...	-0.259238	-0.390576

<표 9> 어절 벡터의 분산표현 예시

	1	2	3	...	98	99
‘영화’	-0.269428	-0.463912	-0.793704	...	-0.615358	0.030695
‘너무’	-0.940717	-0.20902	0.479832	...	-0.052917	0.1087545
‘그냥’	-0.579671	0.1308221	-0.100119	...	0.7588164	-0.260681
‘정말’	-0.526271	0.1496528	0.4064507	...	0.5188472	-0.240719

## 5.2 정확도 및 loss

### 5.2.1 음소 단위 모형의 정확도 및 loss

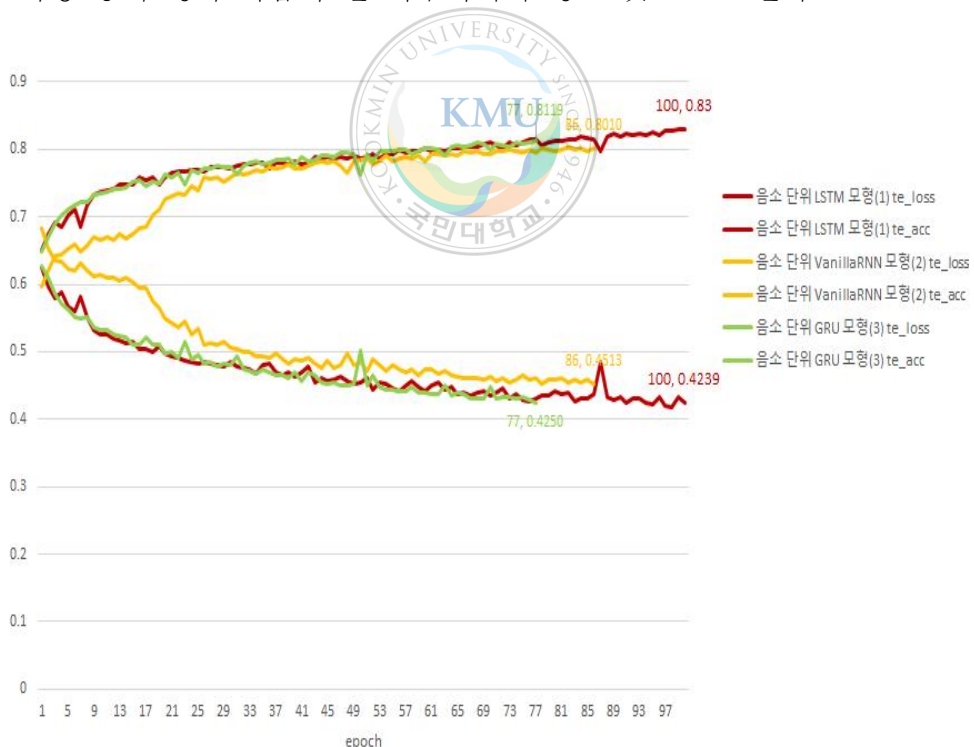
음소 단위 모형의 정확도와 loss를 <표 10>와 같이 정리하였다. tr\_loss와 tr\_acc는 학습을 중단한 epoch에서 학습데이터를 이용해 측정한 loss 및 정확도를 나타내며, te\_loss와 te\_acc는 학습을 중단한 epoch에서 실험 데이터를 이용해 측정한 loss 및 정확도를 의미한다. 또한, training\_epoch는 te\_loss가 더 이상 감소하지 않아 학습을 중지할 때까지의 epoch이다.

<표 10> 음소 단위 모형 결과

	음소 단위 LSTM 모형(1)	음소 단위 Vanilla RNN 모형(2)	음소 단위 GRU 모형(3)
tr_loss	0.2638	0.3753	0.3638
tr_acc	0.888	0.847	0.84
te_loss	<b>0.4239</b>	<b>0.4513</b>	<b>0.425</b>
te_acc	<b>0.83</b>	<b>0.801</b>	<b>0.8119</b>
tr_loss-te_loss (차이)	0.1601	0.076	0.0612
tr_acc-te_acc (차이)	0.058	0.046	0.0281
training_epoch	<b>100</b>	<b>86</b>	<b>77</b>

음소 단위 모형 중 가장 성능이 뛰어난 모형은 LSTM 모형(1)으로 0.83

te\_acc와 0.4239 te\_loss를 갖는다. 또한, 100 epoch 동안 te\_loss가 계속 감소하여 학습을 중지하지 않았다. tr\_loss와 te\_loss 차이, tr\_acc와 te\_acc의 차이는 각각 0.1601와 0.058을 갖는다. 다른 두 모형에 비해 차이가 큰 것은 학습을 오래하여 학습데이터로 측정한 성능이 상승한 것으로 보인다. GRU 모형(3)은 0.8119 te\_acc와 0.425 te\_loss를 가지며 LSTM 모형(1)과 각각 0.0181, 0.0011 차이를 보인다. epoch 77 이후로 te\_loss가 감소하지 않아 학습을 중지하였는데, 3개 음소 단위 모형 중 가장 먼저 학습을 완료하였다. Vanilla RNN 모형(3)은 0.801 te\_acc와 0.4513 te\_loss를 가지며 음소 단위 모형 중 성능이 가장 좋지 않았다. 86 epoch에서 학습을 중지하였고, 학습데이터로 측정한 성능(tr\_acc, tr\_loss)이 가장 좋지 않아 학습이 잘 이루어지지 않은 것으로 보인다.



<그림 16> 음소 단위 모형의 te\_loss 및 te\_acc 변화

위 세 모형의 te\_loss 및 te\_acc를 학습 epoch동안 측정하여 <그림 16>과 같이 나타냈다. 음소 단위 모형들은 모두 안정적인 학습결과를 보이고 있다. LSTM 모형(1)은 epoch 87에서 정확도가 크게 낮아지고 loss가 크게 증가하는 모습을 보였다. 또한, GRU 모형(3)과 비슷한 추세를 보이고 있고 Vanilla RNN 모형(2)은 다른 두 모형에서 비해 epoch 동안 학습이 더딘 모습을 보였다.

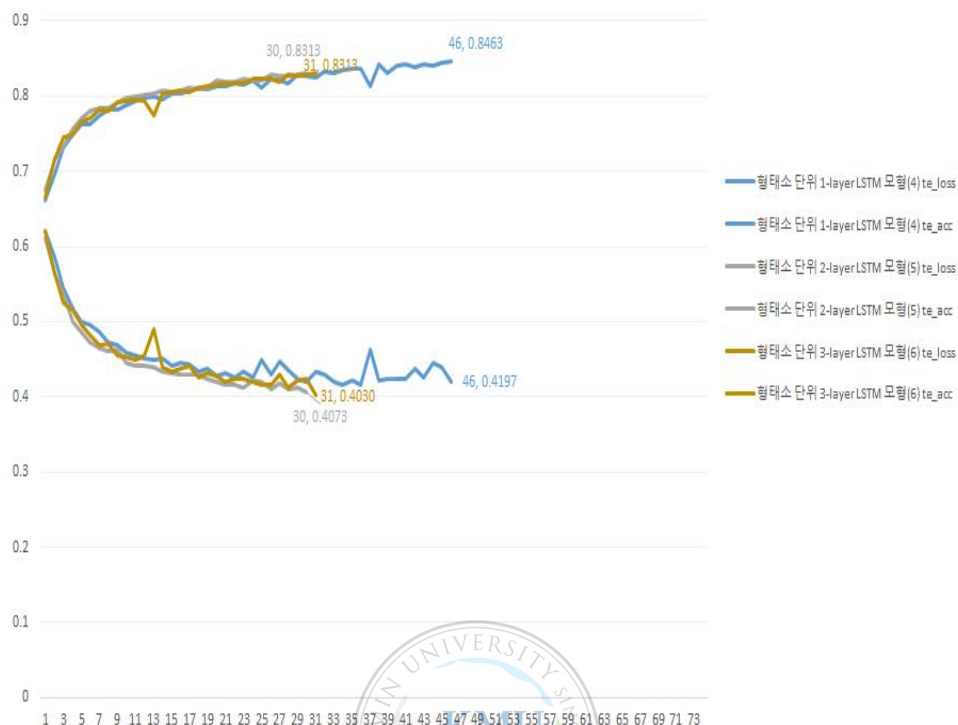
### 5.2.2 형태소 단위 모형의 정확도 및 loss

형태소 단위 모형의 정확도와 loss를 <표 11>와 같이 정리하였다. tr\_loss와 tr\_acc는 학습을 중단한 epoch에서 학습데이터를 이용해 측정한 loss 및 정확도를 나타내며, te\_loss와 te\_acc는 학습을 중단한 epoch에서 실험데이터를 이용해 측정한 loss 및 정확도를 의미한다. 또한, training\_epoch는 te\_loss가 더 이상 감소하지 않아 학습을 중지할 때까지의 epoch이다.

<표 11> 형태소 단위 모형 결과

	형태소 단위 1-layer LSTM 모형(4)	형태소 단위 2-layer LSTM 모형(5)	형태소 단위 3-layer LSTM 모형(6)
tr_loss	0.2306	0.288	0.2901
tr_acc	0.9084	0.88	0.881
te_loss	<b>0.4197</b>	<b>0.4073</b>	<b>0.403</b>
te_acc	<b>0.8463</b>	<b>0.8313</b>	<b>0.8313</b>
tr_loss-te_loss (차이)	0.1894	0.1193	0.1129
tr_acc-te_acc (차이)	0.0621	0.0487	0.0497
training_epoch	<b>46</b>	<b>30</b>	<b>31</b>

1-layer LSTM 모형(4)은 0.8463 te\_acc을 가지며 정확도 측면에서는 가장 성능이 좋았다. 또한, 46 epoch에서 학습을 중지하여 나머지 두 모형에 비해 학습을 오래 지속하여 tr\_loss와 te\_loss차이, tr\_acc와 te\_acc차이가 가장 컸다. 정확도가 가장 높았지만, 0.4197 te\_loss를 가지며 loss 측면에서는 성능이 좋지 않았다. 2-layer LSTM 모형(5)은 0.8313 te\_acc로 3-layer LSTM 모형(6) te\_acc와 같은 값을 가졌지만 0.4073 te\_loss로 3-layer LSTM 모형(6)에 비해 0.0043 큰 값을 가진다. 또한, 30 training\_epoch을 가지며 형태소 단위 모형 중 가장 일찍 학습을 중지하였다. 3-layer LSTM 모형(6)은 정확도 측면에서는 0.8313 te\_acc를 가져 1-layer LSTM 모형(4)과 같지만, loss 측면에서는 0.403 te\_loss를 가지며 3개 모형 중 가장 성능이 좋았다. 또한, 31 training\_epoch를 가져 2-layer LSTM 모형(5)과 1 epoch 차이로 늦게 학습을 중지하였다.



<그림 17> 형태소 단위 모형의 te\_loss 및 te\_acc 변화

위 세 모형의 te\_loss 및 te\_acc를 학습 epoch동안 측정하여 <그림 17>과 같이 나타냈다. 형태소 단위 모형들은 초기부터 안정적인 학습을 진행하였다. 1-layer LSTM 모형(4)은 25 epoch 이후로 정확도와 loss가 급격히 변화하는 불안정한 학습을 보인다. 2-layer LSTM 모형(5)은 일정한 loss 감소 및 정확도 증가형태를 띄우며 형태소 단위 모형 중 가장 안정적인 모습을 보였고, 3-layer LSTM 모형(6)은 11 epoch에서 불안정한 학습을 제외하고 안정적인 학습을 진행하였다.



### 5.2.3 어절 단위 모형의 정확도 및 loss

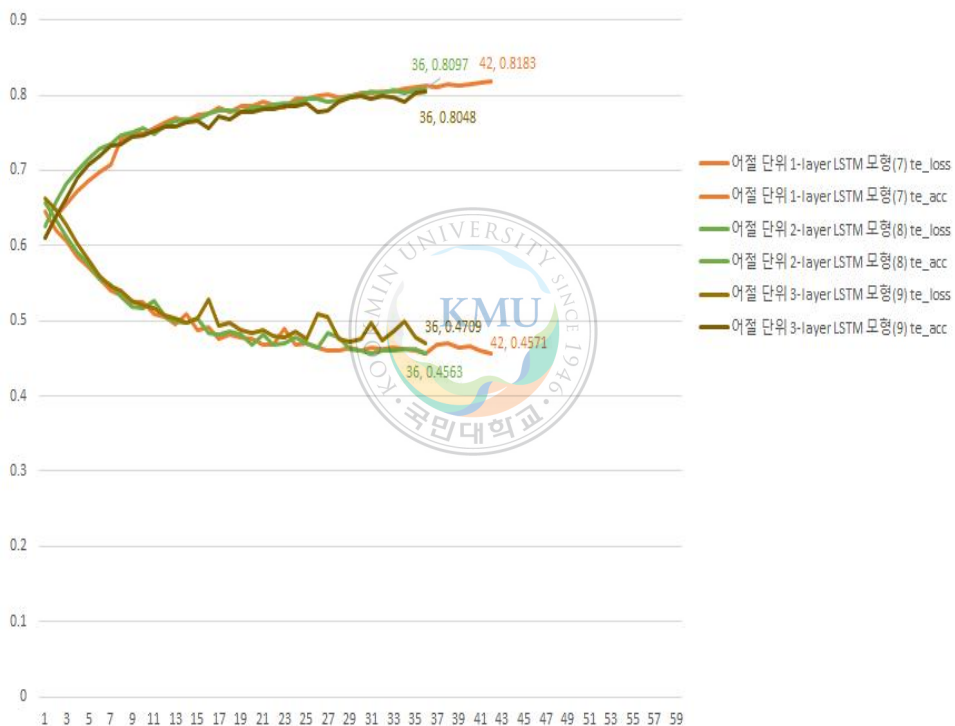
어절 단위 모형의 정확도와 loss를 <표 12>와 같이 정리하였다. tr\_loss와 tr\_acc는 학습을 중단한 epoch에서 학습데이터를 이용해 측정한 loss 및 정확도를 나타내며, te\_loss와 te\_acc는 학습을 중단한 epoch에서 실험 데이터를 이용해 측정한 loss 및 정확도를 의미한다. 또한, training\_epoch는 te\_loss가 더 이상 감소하지 않아 학습을 중지할 때까지의 epoch이다.

<표 12> 어절 단위 모형 결과

	어절 단위 1-layer LSTM 모형(7)	어절 단위 2-layer LSTM 모형(8)	어절 단위 3-layer LSTM 모형(9)
tr_loss	0.3019	0.3225	0.3262
tr_acc	0.8631	0.8533	0.8527
te_loss	<b>0.4571</b>	<b>0.4563</b>	<b>0.4709</b>
te_acc	<b>0.8183</b>	<b>0.8097</b>	<b>0.8048</b>
tr_loss-te_loss (차이)	0.1552	0.1338	0.1447
tr_acc-te_acc (차이)	0.0448	0.0456	0.0479
training_epoch	<b>42</b>	<b>36</b>	<b>36</b>

1-layer LSTM 모형(7)은 0.8183 te\_acc를 가지며 정확도 측면에서 가장 성능이 좋았다. 0.4571 te\_loss를 가지며 2-layer LSTM 모형(8)보다 0.0008 크지만, 큰 차이를 보이지 않아 어절 단위 모형 중 가장 성능이 뛰어난 것으로 보인다. 또한, 42 training\_epoch를 가지며 세 모형 중 가장

늦게 학습을 중지하였다. 2-layer LSTM 모형(8)은 0.4563 te\_loss와 0.8097 te\_acc를 가지며 te\_loss는 가장 낮았다. 또한, 36 training\_epoch를 가지며 3-layer LSTM 모형(9)과 같은 epoch에서 학습을 중지하였다. 3-layer LSTM 모형(9)은 0.4709 te\_loss와 0.8048 te\_acc로 세 모형 중 가장 성능이 좋지 않으며 36 training\_epoch를 갖는다. 위 세 모형의 te\_loss 및 te\_acc를 학습 epoch동안 측정하여 다음 <그림 18>과 같이 나타냈다.



<그림 18> 어절 단위 모형의 te\_loss 및 te\_acc 변화

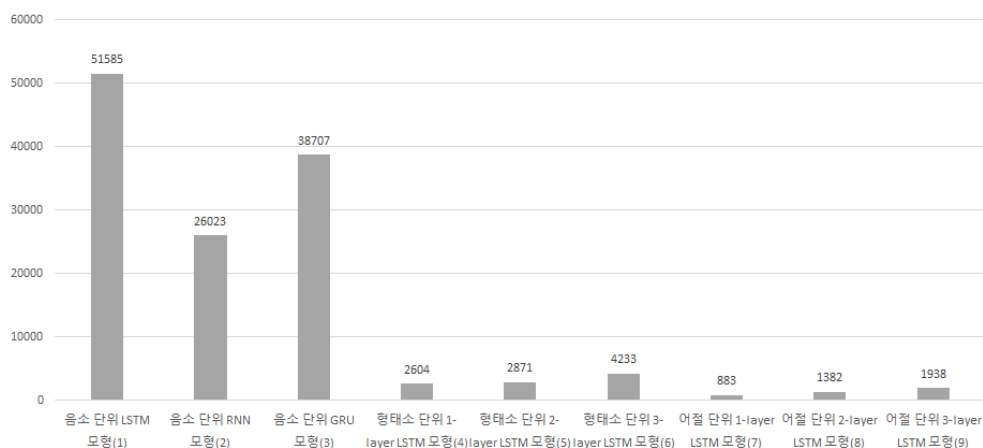
1-layer LSTM 모형(7)과 2-layer LSTM 모형(8)은 안정적인 학습곡선을 보이고 있지만, 3-layer LSTM 모형(9)의 경우 27 epoch이후로 불안정한 학습을 진행하였다. 정확도는 1-layer LSTM 모형(7)이 초반에 학습이

더딘 모습을 보였고 이외에는 2-layer LSTM 모형(8)과 비슷한 학습곡선을 이루고 있다.

### 5.3 학습시간

<표 13> 모형 학습시간

모형	음소 단위 모형			형태소 단위 모형			어절 단위 모형		
	LSTM 모형(1)	RNN 모형(2)	GRU 모형(3)	1-layer LSTM 모형(4)	2-layer LSTM 모형(5)	3-layer LSTM 모형(6)	1-layer LSTM 모형(7)	2-layer LSTM 모형(8)	3-layer LSTM 모형(9)
학습시간	51585 (s)	<b>26023</b> (s)	38707 (s)	<b>2604</b> (s)	<b>2871</b> (s)	4233 (s)	<b>883</b> (s)	1382 (s)	1938 (s)



<그림 19> 모형 학습시간

9개 모형의 학습시간을 <표 13>와 <그림 19>로 정리하였다. 음소 단위 모형에서는 RNN 모형(2)이 26023로 가장 빨랐으며, LSTM 모형(3)은 51585로 가장 느렸다. 형태소 단위 모형에서는 1-layer LSTM 모형(4)이 2604 으로 가장 빠르며 3-layer LSTM 모형(6)이 4233 학습시간을 가지며 가장 느렸다. 어절 단위 모형에서도 1-layer LSTM 모형(7)이 883 학습시간으로 가장 빠르며, 3-layer LSTM 모형(9)이 학습시간 1938로 가장 느렸다. 전체적으로 음소 단위 모형이 가장 오래 학습을 진행하였고, 어절 단위 모형 학습이 가장 짧았다.



## 제6장 결론 및 향후 연구계획

본 논문에서는 사람들이 인터넷에 작성한 영화 후기 데이터를 대상으로 한국어 문장을 수치화하고 감성사전을 구축하지 않는 감성 분석 모델을 설계하는 방법을 다루었으며, 한국어를 다양한 방법으로 분할하고 RNN을 이용해 영화 후기 문장의 감성이 긍정인지 부정인지를 예측하는 모델을 구축하였다. 총 9개 실험을 진행하여 정확도, loss, 학습시간을 비교하였다. 정확도는 모두 80% 이상의 정확도를 가졌으며, 음소 단위 모델을 제외한 형태소 및 어절 단위 모형들은 정해진 100 epoch 이전에 학습을 완료하였다.

입력 단위별 모형 특징을 살펴보면, 음소 단위 모형은 LSTM 모형성능이 가장 뛰어났다. 특히, LSTM 모형은 실험 기준인 100 epoch동안 꾸준히 loss가 감소하였기 때문에 더 많은 epoch로 학습한다면 성능을 높일 수 있을 것으로 보인다. 학습시간은 RNN 모형이 가장 빨랐으며 LSTM 모형이 가장 느렸다. 형태소 및 어절 단위 모형과 비교하면 대체로 어절 단위 모형보다 성능이 뛰어나지만, 형태소 단위 모형에 비해 성능이 떨어졌다. 또한, 문장을 가장 작은 단위로 분할했기 때문에 문장길이가 길어져 형태소 및 어절 단위 모형에 비해 학습시간이 길어졌다. 이러한 특징을 가지는 음소 단위 모형은 문장 벡터를 구성할 때 음소만 포함하지 않고 감성을 나타낼 수 있는 특수문자 등을 포함한다면, 성능을 높일 수 있을 것으로 보인다.

형태소 단위 모형은 1-layer LSTM 모형이 가장 뛰어났다. 학습시간은

LSTM 층 개수로 인해 학습해야 할 가중치 및 편향이 상대적으로 적은 1-layer LSTM 모형이 가장 작았다. 또한, LSTM layer 수는 모형 성능에 영향을 주지 않았다. 음소 및 어절 단위 모형과 비교하면 모든 모형의 정확도 및 loss 성능이 뛰어남을 확인했다. 학습시간은 어절 단위 모형에 비해 큰데, 이는 형태소로 분할한 문장길이가 어절 단위로 분할한 문장길이보다 큰 것으로 보인다. 이러한 특징을 가지는 형태소 단위 모형은 데이터를 구축 시, 사전크기와 단어벡터의 차원을 다양하게 설정한다면 모형 성능을 개선할 수 있을 것으로 보인다.

어절 단위 모형에서는 1-layer LSTM 모형 성능이 가장 뛰어났다. 형태소 단위 모형과 마찬가지로 LSTM layer가 증가할수록 학습속도가 느려졌으며 layer 수는 모형 성능에 영향을 미치지 않았다. 음소 및 형태소 단위 모형과 비교하면 학습시간이 작운데, 이는 한 문장안의 어절 수가 가장 적은 것으로 보인다. 또한, 어절 단위 모든 모형의 성능이 음소 및 형태소 단위 모형에 비해 성능이 좋지 않았다. 이러한 어절 단위 모형은 형태소 단위 모형과 마찬가지로 데이터 구축 시, 사전크기와 단어벡터의 차원을 다양하게 설정한다면 모형 성능을 개선할 수 있을 것으로 보인다.

위 9개 실험의 한계점으로는 수집한 영화 후기 문장을 살펴보면 긍정 또는 부정으로 나눈 기준의 모호성이 존재하였다. 예로 부정으로 분류한 별 5개 후기가 긍정을 내포하고 있었으며 긍정으로 분류한 별 10개 후기가 부정을 내포하고 있어, 정확한 데이터 분류가 이루어져야 한다. 또한, 영화 후기 뿐 아니라 상품과 콜센터 등 다양한 후기 데이터를 이용해 학습하여 도메인에 영향을 받지 않는 모형 설계가 필요하다.

본 논문에서는 한국어 문장을 다양한 방법으로 수치화해 감성을 예측하는 실험을 진행하였고, 이는 영어에 비해 연구가 부족한 한글 기반 언어 처리 및 음성인식 분야에 활용될 것으로 기대한다. 향후 연구로는 음소를 임의의 수치로 설정하지 않고 Word2vec과 같이 수치화하는 모델을 설계해 적용하거나, RNN과 함께 기존 감성분석 연구의 핵심 중 하나인 감성 사전에 활용한다면 보다 뛰어난 성능을 지닌 감성분석 모델을 구축할 수 있을 것으로 보인다.



## 참고문헌

권수정, “word2vec과 rnn을 이용한 영화 리뷰의 감성분석”, 동국대학교, 2017.

김유영, 송민, “영화 리뷰 감성분석을 위한 텍스트 마이닝 기반 감성 분류기 구축”, *지능정보연구*, 제22권, 제3호, 2016, 71-89.

김양훈, 황용근, 강태관, 정교민, “LSTM 언어모델 기반 한국어 문장 생성”, *한국통신학회논문지*, 제41권, 제5호, 2016, 592-601.

마크로밀 엠브레인, “소비자 리뷰 영향력 조사”, 2014.

안성만, 정여진, 이재준, 양지현, “한국어 음소 단위 LSTM 언어모델을 이용한 문장 생성”, *지능정보연구*, 제23권, 제2호, 2017, 71-88.

한남기, “word2vec 학습 자료를 사용한 새로운 한글 개체명 인식 모델 제안”, 연세대학교, 2016.

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, ... and M. Kudlur, “TensorFlow: A System for Large-Scale Machine Learning.”, *OSDI*, Vol.16, 2016, 265-283.

Appel, O., F. Chiclana and J. Carter, “Main concepts, state of the art



and future research questions in sentiment analysis.”, *Acta Polytechnica Hungarica*, Vol.12, No.3, 2015, 87–108.

Bengio, Y., P. Frasconi and P. Simard, “The problem of learning long-term dependencies in recurrent networks.”, *IEEE International Conference on Neural Networks, San Francisco, US*, 1993, 1183–1188.

Cauchy, A., “M’ethode g’en’erale pour la resolution des systemes d’equations simulta’nees.” *Comp. Rend. Sci, FR*, 1847, 536–538.

Cho, K.H., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation.”, *arXiv preprint arXiv:1406.1078*, 2014.

Chung, J.Y., C. Gulcehre, K.H. Cho and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.”, *arXiv preprint arXiv:1412.3555*, 2014.

Church, K.W. and P. Hanks, “Word association norms, mutual information, and lexicography.”, *Computational linguistics*, Vol.16, No.1, 1990, 22–29.

Feldman, R., “Techniques and Applications for Sentiment Analysis.”, *Communications of the ACM*, Vol.56, No.4, 2013, 82–89.

Ferguson, P., N. O'Hare, M. Davy, A. Bermingham, P. Sheridan, C. Gurrin, and A.F. Smeaton, "Exploring the use of paragraph-level annotations for sentiment analysis of financial blogs.", *Proceedings of WOMAS 2009-Workshop on Opinion Mining and Sentiment Analysis, Seville, Spain*, 2009.

Goller, C. and A. Kuchler, "Learning task-dependent distributed representations by backpropagation through structure.", *IEEE International Conference on Neural Networks, Washington, US*, 1996, 347-352.

Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, The MIT Press, 2016.

Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory.", *Neural Computation*, Vol.9, No. 8, 1997, 1735-1780.

Hu, Z., J. Hu, W. Ding, and X. Zheng, "Review sentiment analysis based on deep learning.", *Proceedings of e-Business Engineering (ICEBE) IEEE 12th International Conference*, 2015, 87-94.

Irsoy, O. and C. Cardie, "Opinion mining with deep recurrent neural networks.", *In Proceedings of the 2014 conference on empirical methods in natural language processing, Doha, Qatar*, 2014, 720-728.

Jin, W., H.H. Ho, and R.K. Srihari, "OpinionMiner: a novel machine learning system for web opinion mining and extraction.", *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, FR*, 2009, 1195–1204.

John A.H., "Online shopping.", Pew Internet & American Life Project Report, 2008.

Jozefowicz, R., W. Zaremba and I. Sutskever, "An empirical exploration of recurrent network architectures.", *ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, FR*, 2015, 2342–2350.

Kim, Y., "Convolutional neural networks for sentence classification.", *arXiv preprint arXiv:1408.5882*, 2014.

Kingma, D. and J. Ba, "Adam: A Method for Stochastic Optimization." *arXiv:1412.6980*, 2014.

Lankinen, M., H. Heikinheimo, P. Takala, and T. Raiko, "A Character-Word Compositional Neural Language Model for Finnish." *arXiv Preprint arXiv:1612.03266*, 2016.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based

learning applied to document recognition.”, *Proceedings of the IEEE*, Vol.86, No.11, 1998, 2278-2324.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space.”, *arXiv preprint arXiv:1301.3781*, 2013.

Olah, C., “Understanding LSTM Networks.”, 2015. Available at <http://colah.github.com.io/posts/2015-08-Understanding-LSTMs>(Accessed December 1. 2017).

Pak, A. and P. Paroubek, “Twitter as a corpus for sentiment analysis and opinion mining.”, *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, 2010, 1320-1326.

Pang, B. and L. Lillian, “Opinion mining and sentiment analysis.”, *Foundations and Trends® in Information Retrieval*, Vol.2, No.1-2, 2008, 1-135.

Pang, B. and L. Lillian, Sentiment Analysis and Subjectivity, *Handbook of natural language processing*, 2010.

Pang, B., L. Lillian and V. Shivakumar, “Thumbs up?: sentiment classification using machine learning techniques.”, *Proceedings of the ACL-02 conference on Empirical methods in natural language*

*processing*, Vol.10, 2002, 79–86. doi:10.3115/1118693.1118704 (Download February 27, 2017)

Scaffidi, C., K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin, “Red Opal: product–feature scoring from reviews.”, *In Proceedings of the 8th ACM conference on Electronic commerce, US*, 2007, 182–191.

Schuster, M. and K.K. Paliwal, “Bidirectional recurrent neural networks.”, *IEEE Transactions on Signal Processing*, Vol.45, No.11, 1997, 2673–2681.

Socher, R., A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A. Ng, and C. Potts, “Recursive Deep Models for Semantic Compositionality over A Sentiment Treebank.”, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, US*, 2013, 1631–1642.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutslever, and R. Salakhutdinov, “Dropout :A simple Way to Prevent Neural Networks from Overfitting.”, *Journal of Machine Learning Research*, Vol.15, No.1, 2014, 1929–1958.

Taboada, M., J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based Methods for Sentiment Analysis.”, *Computational Linguistics*, Vol.37, No.2, 2011, 267–307.

Turney, P.D., "Thumbs up or Thumbs Down? : Semantic Orientation Applied to Unsupervised Classification of Reviews.", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, US*, 2002, 417-424.

Zhu, F. and X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics.", *Journal of Marketing*, Vol.74, No.2, 2010, 133-148.



## Abstract

# Sentiment Analysis based on Korean using Recurrent Neural Network: focused on Online Movie Review

*by JaeJun Lee*

Department of Data Science  
Graduate School, Kookmin University,  
Seoul, Korea



Internet development makes access easy to the online review which contained opinions and experiences of others. Reviews can be utilized as an important marketing strategy for the company due to their actual influence on the product and service.

Sentiment analysis research was accomplished to sensor the subjective information like opinion and attitude in the words. Research was made by sentiment dictionary in the beginning, but glowing sentiment analysis research recently based on deep learning.

Language model, together with sentiment analysis, which can expect next words or sentences by studying corpus is using deep learning. In language model experiment using Korean sentences, various corpus is applied but sentiment analysis with morpheme only.

This study implement sentiment analysis to distinguish negative or positive using various Korean corpus. It forecast sentiment not by sentiment dictionary construction used current sentiment but by corpus only composed sentences. By construction for morpheme-level model and phoneme-level model which is the smallest unit in Korean and word-level model, analyze model performance by input unit.

Phoneme-level model was constructed 3 RNN layers like LSTM, Vanilla RNN, GRU. After morpheme and word-level model expressed dispersion and constructed the models which had 2 or 3 layers included. Measured accuracy and loss for performance evaluation and compared learning hours per model.

By experiment result, phoneme-level model was standing high above the others and confirmed not effect LSTM layer for model performance in morpheme and word-level model. Learning hours is the largest in phoneme-level model which the sentence becomes longer by divided in the smallest unit and the smallest in word-level model.

Through sentiment analysis experiment for the Korean sentences by the various corpus, expect implemented in language processing and speech recognition which have lack of research compared to English.