CNN – BILSTM 조합모델을 이용한 악성댓글 분류 분석

국민대학교 소프트웨어전공 홍성표

인터넷 기술과 소셜 미디어의 빠른 성장으로 비정형 문서 표현도 다양한 응용 프로그램에서 사용 가능한 마이닝 기술이 발전되었다. 특히 댓글 분석은 제품이나 서비스에 내재된 사용자의 감성 및 의도를 탐지 할 수 있는 분석방법으로 지난 몇 년 동안 많은 관심을 받고 있다. 초기의 댓글 분석 연구에서는 간단한 리뷰 중심으로 진행되었지만, 최근에는 블로그, 뉴스기사, 영화 리뷰, SNS, 등 다양한 분야에 적용되고 있다. 이는 사람들의 감성 및 의도 공격성 등 긍정 및 부정의 범주로 구분 분석하고 있다. 현재까지 댓글 분석에 대하여 많은 분석에 진행되어 왔으나 대부분 전통적인 단일 기계학습기법에 의존한 분류이기 때문에 정확도가 떨어진다. 본 연구에서는 전통적인 기계학습기법 대신 딥러닝을 이용한 분류 정확도를 사용한다. 따라서 딥러닝 모델을 조합 및 활용하여 긍정, 부정에 해당하는 분류 정확도를 개선한다. 본 연구에서는 다양한 딥러닝의 모델을 설계하여 실험하였으며 그 중 CNN-LSTM 조합 모델의 성과가 가장 우수한 것으로 나타났다.

---

1. 서론

전통적으로 정형 데이터에 적용되어 왔으나 최근 텍스트, 이미지, 오디오 데이터 등 많은 연구가 진행되고 있다. 그 중, 텍스트 분석을 다루는 NLP(Natural Language Processing) 연구가 활발 하며, NLP 연구는 온라인 텍스트 감성, 주관적인 생각, 감정 등을 식별하고 서비스에 대한 느낌을 분석할 수 있다. 즉 텍스트 시퀀스를 사전 정의된 범주를 이용하여 분류하는 분석방법이다. 감성 분석 중 문장 관점에서의 분류 분석을 극성 분석이라 하며, 극성 분석은 문서 내 의견이 긍정인지, 부정인지, 중립인지를 분류하는 것이다. 최근 댓글과 같은 비정형 데이터의 댓글의 성향 분류는 정형 데이터 보다 더 정교한 분류를 가능하게 하고 댓글 내의 문장의 순서 등을 고려할 수 있는 기법이 반드시 필요하다. 선행 연구에서는 주로 인공신경망, SVM 등 단순 기계학습법을 활용하여 왔으나, 텍스트의 불완전성, 오탈자 등 여러가지 이유 때문에 분류 성과가 미흡한 경우가 많았다. 본 연구에서는 이러한 한계점을 보안하고자 최근에 기계학습 분야에서 활발하게 연구되고 있는 대표적인 딥러닝 기반 자연어처리를 위한 방법으로 CNN-BILSTM 모델에 대하여 연구하였다. CNN은 주로 얼굴인식이나 이미지 분류에서 사용되는 알고리즘이지만, NLP에서 Bag of words와 유사하게 사용된다. 또한 LSTM은 주어진 단어를 미리 예측할 수 있도록 순차적으로 배열 할 수 있는 장점이 있어 챗봇과 텍스트 번역에서 유용하게 이용되는 알고리즘이다. 따라서 CNN-LSTM의 조합 모델을 이용하게 되면 양자의 장점을 활용할 수 있으며, 이

를 통해 댓글 분석의 성능을 개선할 수 있다. 한편, 딥러닝 기반 모델들은 NLP 등 여러분야에서 전통적인 기계학습기법보다 뛰어난 성능을 갖고 잇는 것을 알려져 있다. 하지만 실제 응용에 있어서는 과적합이 되지 않도록 하는 것이 중요하다. 특히, 딥러닝 기법은 이미지 인식 등에서 활발하게 이용되고 있으나, 댓글 분석 등의 텍스트 분석 사례는 적기 때문에 본 연구에서 댓글 데이터의 극성 분류 문제에 이용해 보고자 한다. 본 연구의 순서는 다음과 같다. 2장에서는 댓글 분석과 관련된 선행연구를 소개하고, 3장에서는 제안하는 CNN-LSTM 조합 모델이 어떻게 구현되는지를 설명한다. 4장에서는 논문의 분석 과정과 결과를 설명한다.

2. 선행연구

댓글 분석은 텍스트에서 상황 별 속성에 대한 주관적인 의견과 감정을 분류하고 식별하는 분석 방법이다. 기본적인 분석에는 긍정과 부정에 대한 단어 빈도수를 기반으로 제한되지만, 댓글 분석을 기계학습기반의 분류 문제로 모델링 할 경우 사용자의 의도와 반응을 도출할 수 있다.

자연어 분석에서 인공 신경망, SVM, 최대 엔트로피 등이 댓글 분석에 많이 활용되는 기법이다. 댓글 분석 데이터 셋에 n-gram, TF-IDF,BOW에 활용하여 댓글 분석을 진행하였으며 SVM과 최대 엔트로피, 인공 신경망(단순신경망) SVM 모델이 가장 성능이 높았다.

3. 데이터 셋

3.1 데이터 셋의 구성

본 데이터 셋은 인공지능 경진대회의 데이터 셋 구성으로 이루어져 있다.데이터 셋의 샘플은 다음 <표1>과 같다

.

댓글	bias	Hate
쓰레기 뉴스임	None	Hate
여자는 얼굴보다 요리 실력 근데 얼굴도 이쁘심	Gender	None
강한나는 좋은기사 올라온게 거의 없네	None	None
세븐 걸린거 넌 잘피해갓지 다 그놈이 그놈인데	Gender	offensive
좌파 조국은 잘못 절대 인정안하는데 우파는 쪽팔린거 알아서 잘못은 인정하지 좌파놈들 역겹다.	others	hate

<표1>

데이터를 크게 먼저 2가지 분류를 진행하였다. 댓글에 대한 편견(bias), 공격적인 정도(hate)를 나누고 이후에 세부적으로 카테고리를 나누었다. 먼저, 편견 부분에서는 다시 3개의 카테고리로 나누어져 있다. 첫째로, gender 성적 지향성, 성 정체성, 성별에 따른 역할이나 능력에 대한 편견 내용이 담겨있다. 두번째로는 others 성별 외 인종이나 출신 지역, 피부색 종교, 장애, 직업 등에 대한 편견 none 편견 존재 하지 않음. 공격 정도 hate 또한 3개의 카테고리로 세분화 하여 진행한다. Hate: 대상을 심하게 비난하거나 깍아 내려서 정신적 고통 등을 야기할 수 있는 표현 두번째로는 Offensive 모욕이나 혐오에는 미치지 않지만, 공격적이고 무례한 내용, 마지막으로 none 모욕이나 공격성이 존재하지 않음을 나타낸다.

이렇게 총 train_data 는 7787개 이며 validation data 총 500 문장으로 이루어져 있으며 데이터 전처리를 통하여 학습을 진행하였다.

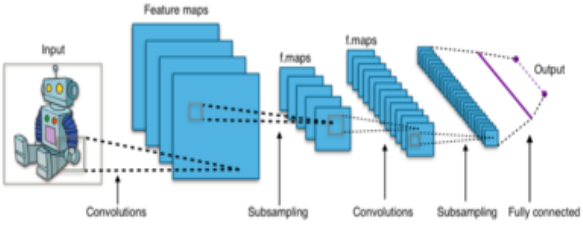
3.2 데이터 전처리

데이터 전처리 과정은 크게 4가지 다음과 같은 방법이었다. 데이터 전처리를 하는 이유는 말뭉치를 학습하는데 있어서 기본이 되는 사전을 만들기 위한 작업이다. 데이터 전처리 방법은 다음과 같이 진행한다. 전처리 방법은 다음과 같다. 첫번쨰로 단어를 한국어 형태소 분석 방법인 SoyNlp를 사용하여 단어를 형태소 분리 한다. 여기서 형태소를 분리 진행할 때는 문장의 규칙을 나타내어 형태소 분석한다. 예를 들어 "나는 2등도 못하는 바보 똥개 shit"이야 라는 문장이 있으면 다음과 같이 형태소 분석한다. '나', '는', '1등', '도', '못하'는, '바보', '똥', '개', 'shit'으로 분류한다. 그 이후 단어 사전을 만들기 위한 방법을 진행한다. 댓글 같은 경우는 한 사람이 작성하는 것이 아닌 여러 사람이 작성하므로 개인마다 말투, 성향이 모두 다르기 때문에 모든 단어를 사전으로 만들게 되면 크게 의미 없는 단어가 담길 수 있다. 예를 들면 연예인 이름, 고유 명사, 은어, 같은 경우는 많이 등장하지 않기 때문에 이러한 단어는 사전에서 제거 하기 위해 평균 3번 이하로 나온 단어는 단어장에서 제거하는 전처리를 거친다. 그렇게 되면 단어 사전을 만들게 된다. 그 이후 단어를 정수화 하는 과정이 필요하다. 단어 사전에 있는 말들을 정수로 표현 해준다. 마지막으로 모든 문장들을 하나의 문장처럼 만들기 위해 단어장에서 나오는 가장 긴 단어의 길이로 모두 통일하는 작업을 진행한다. 위와 같은 작업을 Padding이라고 한다. Padding을 진행하게 되면 모든 문장의 길이는 50으로 일정하게 되며 모든 데이터의 전처리를 마친다.

3. CNN-LSTM 조합 모델

3.1 딥러닝 기법

딥러닝은 일반적으로 기계학습법의 하나인 신경망의 계층을 심화시킨 알고리즘을 의미한다. 전통적인 기계학습기법은 분류에 대한 특징 집합을 따로 추출하여야 하지만 딥러닝은 특징 집합을 추출하는 과정뿐만 아니라, 복합한 특징을 자동으로 추출할 수 있다. 딥러닝 기법 중 대표적인 기법인 CNN(Convolutional Neural Network)은 convolution layer와 pooling layer로 구성되어 있는데, convolution layer는 가중치와 편향을 적용하고, pooling layer는 활성화를 수행한 convolution layer에서 벡터를 형성하여 값을 가져온다. CNN은 오탈자가 있어도 댓글 분석에 탁월하기 때문에 사전 없이 사용되기도 한다. <그림1>은 일반적인 CNN 구조를 그림으로 표현한 것이다.



<그림1> CNN Structure

한편, LSTM(Long Short-Term memory)은 장기 의존성 문제를 해결하기 위해 고안된 모델로, 입력 게이트, 출력 게이트, 망각 게이트 등 3개의 gate 구조를 가지고 있다. 입출력 시점의 형태를 LSTM에 전달하면, 현재 시점의 출력이 전 연결 계층에 전달되어 업데이트 된 시점을 출력할 수 있게 된다. 본 연구에서는 분류 모델의 성능을 항상 시키기 위해 RNN 일종인 LSTM 이용하여 댓글 분석 정보를 추출하고, 추출된 댓글 분석의 공격성 및 사용자의 감정을 분석하여 모델 성능 평가를 실시한다.

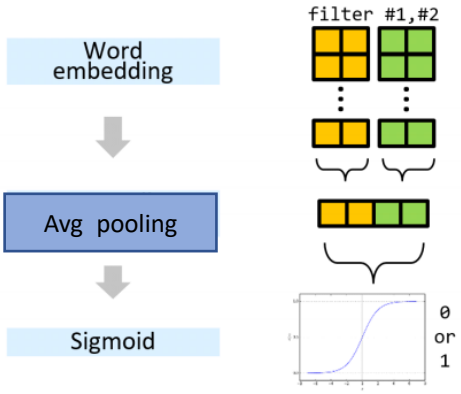
가장 성능이 좋은 모델은 댓글 분석의 편견 및 공격성의 특징을 추출하기 위해 CNN과 LSTM으로 이루어진 모델이다. 기존 기계 학습과 달리 CNN은 convolution layer를 적용하여 특징의 자동 추출이 가능하고, 대규모 병렬 처리가 가능하다. 한편, LSTM은 CNN과 달리 대

규모 병렬 처리가 가능하지 않지만, RNN과 다르게 원하는 시기에 진행 및 제어 할 수 있는 입력, 출력, 망각 게이트가 있다. LSTM을 CNN의 Pooling layer에서 이용할 경우, end-to-end 구조를 띄기 때문에 공간 및 시간적인 특징을 동시에 고려할 수 있다. 그 뿐만 아니라, LSTM은 단어를 예측할 때, 동일하게 sequence 벡터를 모델링할 수 있기 때문에 정확도를 향상시킬 수 있다. 먼저 word embedding을 수행하는데 이는 단어를 표현하기 위해서 사용되는 NLP 작업 중 하나 이며, 댓글 분석에서 문장 간 유사성과 bias, hate의 레이블을 훈련시킨다. Word Embedding의 출력을 통해 텍스트 매트릭스 벡터를 만드는 과정에서 CNN을 도입한다. CNN의 convolution layer의 목적은 kernel의 크기를 조정하여 다른 단어와 함께 사용될 때 다른 단어를 이해하게 할 수 있는 구조가 되게 하는 것이다. 이 구조를 이용하여 CNN이 쉽게 지역 특징을 추출할 수 있다. CNN에서는 각 단계마다 그래디언트 소멸 문제를 완화하기 위하여 ReLU를 활성화함수로 이용한다. 각 과정에서 ReLU를 진행 한 후, 데이터 중 전체 단어로 구성되어 있는 부분을 CNN이 인지하고 Pooling 의 한 방법으로 단일출력으로 변환하여 지정한 영역의 값들을 한 곳으로 모을 수 있도록 하는 역할을 한다. Avg pooling 이후에는 과적합을 방지하고 특정 입력에 집중하지 않도록 하기 위해 레이어의 입력 일부를 임의로 설정하는 dropout 과정을 거친다. LSTM layer의 경우, CNN의 장기 의존성 문제를 보완하기 위하여 이용된다. LSTM은 순차 데이터 특성으로 사용되며, cell state 라는 변수를 두어 메모리에 기억한다. LSTM은 입력, 출력, 망각 게이트를 갖고 있기 때문에 가변적으로 데이터 상황에 따라 제어할 수 있다. 다음으로 완전히 연결된 계층에서 단일 값을 출력한다.

4. 모델 설계

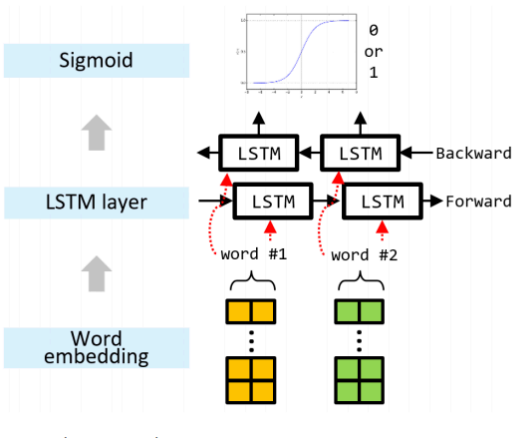
기존 선행 연구 중 가장 성능이 좋았던 SVM과 제안 모델을 통하여 분석을 진행하였다. 먼저, 비교모형으로 사용하기 위하여 댓글 분석과 유사한 텍스트 분류 알고리즘에서 높은 수준의 분류 성능을 보이는 기계학습기법인 SVM을 선정하였고 전통적인 기계학습기법에서 사용하기 위해서 텍스트를 숫자로 벡터화해야한다. 벡터화 하는 방법은 Bag Of Word, TF-IDF 2가지 방법을 사용하였다. BOW(bag of words)는 텍스트 문서의 문법과 단어 순서를 무시하여 코퍼스 전체를 기반으로 어휘 목록을 작성한다. 이 어휘목록들은 각각 숫자 벡터로 표기되며, 데이터 확장성과 시계열 분류 텍스트 처리 등 여러 영역에서 효율적으로 모델링 할 수 있으며, 이미지와 관련된 연구에서도 많이 활용되고 있다. 현재는 비정형 데이터의 기계학습이나 딥러닝 등을 이용한 분류문제에서 새로운 접근법을 제시할 때 쓰이고 있다. TF-IDF(Ter Frequency-Inverse Document Frequency)는 각 문서에 대한 상대 용어 빈도를 이용하는 방법으로 TF는 문서 내의 단어의 빈도수를 나타내며, IDF는 동일한 단어 내에서의 또 다른 문서별 빈도수를 정의한다. TF-IDF 가중치가 높은 단어는 TF-IDF 가중치가 낮은 단어보다 중요하며, 가중치가 높을 수록 문서의 단어 의미 중요도가 높다. 본 연구에서는 데이터 전처리 후 텍스트를 벡터로 변환할 때, BOW, TF-IDF를 사용하여 문서에 대한 특징을 추출하여 알고리즘을 비교할 수 있도록 하였다. 두 번째, 딥러닝의 대표적인 기법인 LSTM을 단일 비교 모형으로 채택했다. 세 번째, 성능이 가장 CNN-LSTM 조합한 모델을 제시한다. CNN-LSTM 조합 모델은 각 기법의 약점을 서로 보안할 수 있고 특히 LSTM을 조합한 모형을 제시한다. 특히 LSTM의 end-to-end를 이용하여 layer 별로 학습 성능을 향상시킬 수 있는 장점이 있다. 딥러닝 분석을 위해, lemmatization된 텍스트를 대상으로, 단어별

토큰화를 다시 진행하였다. 단어별 토큰화를 진행하면, 데이터를 벡터처럼 숫자로 변화할 수 있을 뿐만 아니라 CNN과 LSTM의 딥러닝에 용이하게 사용할 수 있는 sequence 데이터로 만들 수 있게 된다. 이 분석에서 형태소로 전처리된 문장의 최대 길이는 4443 이기 때문에 댓글 문장의 길이 차를 고려하여 단어 벡터의 최대 길이는 4500으로 재구성하여 word embedding 과정을 진행하였다. Dimenstion은 100으로 구성하였다. 4500 x 100 으로 구성된 word embedding을 토대로 convolution layer를 구성하였다. CNN 층은 그림 2와 같다. Convolution layer는 filter를 1개씩 할당하는 Convolution layer 1D 3개로 구성하였으며, kernel은 n-gram 효과를 주기 위하여 3,2,1 로 진행하였다. 활성화함수는 Relu, stride는 1로 설정하였다. 전역으로 max pooling layer를 사용하여 Kernel node 중 최댓값을 구하도록 하였다. Feature map을 설정하기 위하여 256개의 filter에 활성화 함수를 Relu를 적용한 후, 1개씩 Filter에 활성화 함수 Sigmod를 적용하여 분류기 함수에 근사 값을 갖게 하였다. 학습률을 설정할 수 있는 알고리즘으로 Adam(Adaptive moment), SGD를 이용하여 속도 벡터와 그레디언트 누적 벡터를 계산한 후 다음단계에서 정확도를 측정할 수 있도록 하였다. Epoch은 100, batch size를 64로 설정하여 정확도를 측정하였다. 전체적 구조는 <그림2>와 같다.



<그림2>

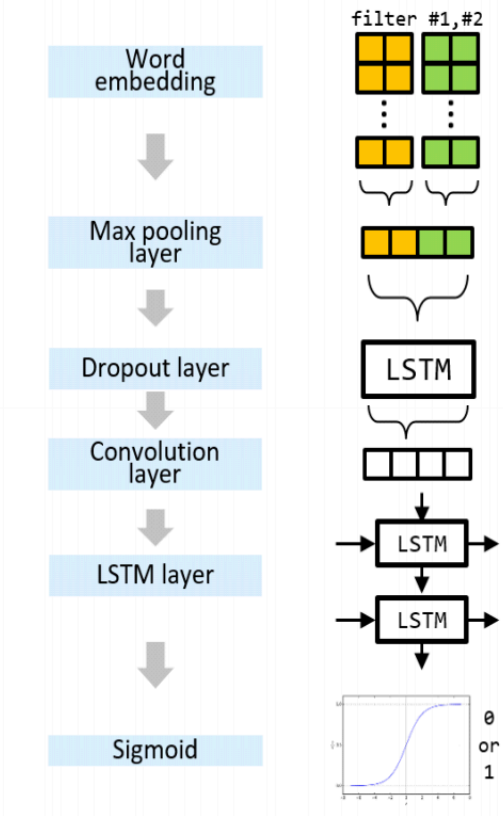
한편, LSTM만 적용한 경우에는 bi-LSTM과 비교할 수 있도록 word emedding 값을 동일하게 설정한 후 진행하였다. 최대 길이는 4500, dimension은 100으로 구성한 후 , 양방향 LSTM Layer를 구성하였다. 이 과정은 <그림3>과 같다.



<그림3>

LSTM layer의 설정은 sequence 분류 문제에서 모델 성능을 향상할 수 있도록, 입력 게이트는 forward로 설정하고 출력 게이트는 backward로 설정하였다. 단방향보다 양방향으로 설정한 이유는 sequence가 진행될 때, 더 빠른 속도로 완전한 학습이 가능하기 때문이다. 128개의 뉴런을 앙뱡향 LSTM layer로 생선한 후 , 25%의 dropout layer를 삽입하여 LSTM 반복학습을 유지할 수 있도록 지정하였다. Dropout layer 이후에는 CNN처럼 감성을 0-9로 모델링 할 수 있도록 sigmoid를 활성함수로 이용하였다.

본 연구의 제안 모형인 CNN-LSTM의 조합모델의 경우는 CNN, LSTM의 단일모델과 같이 최대길이는 4500, dimension 은 100으로 word embedding 값을 동일하게 설정하였다. CNN으로 설정할 때는 filter는 3개, kernel size는 3,2,1 feature map을 구성하기 위한 pool size를 2로 구성한 뒤, LSTM과 동일하게 25%의 dropout layer를 생성하였다, 이후에는 Adam, SGD으로 진행하였다. 이 과정은 <그림4>과 같다.



<그림4>

5. 검증방법

모델 설계에 있어서 검증 방법은 다음과 같다. Test 데이터 500문장을 준비하였다. Test 데이터는 한쪽으로 치우치지 않게 하기 위해 편향성과 공격성이 둘다 None, none인 데이터의 수는 약 30% 정도만 두고 나머지는 골고루 분포한 데이터를 준비하였다.. 그 이후 본 연구에서 설계한 모델의 예측 값을 가지고 test 데이터의 정답 라벨과 비교하여 검증의 정확도를 측정한다. 라벨이 맞는 수를 카운트하여 벡분율로 바꿔 표현한다.

6. 결론 및 결과 도출

6.1 결과 도출

Training	vector	Precision
SVM	TF-IDF	0.55
	BOW	0.52
LSTM	Word Eembedding	0.44
BI-LSTM	Word Eembedding	0.48
CNN+BILSTM	Word Eembedding	0.88

전통적인 단일 기계학습법에서 BOW, TF-IDF는 정학도 0.5 정도를 보여주었다. 기계학습법 중 SVM 기반의 분류 모델이 Bow와 TF-IDF가 단일 딥러닝 모델보다도 높은 정확도를 보여주고 있다. 이 결과는 댓글 분석의 편향성과 공격성을 판단하는 모델이기 때문에 모델의 정확도 측면에서 낮을 수 밖에 없다. 예를 들어서 ("남자친구" 박보검, 송혜교, 진심어린 고백" 내안에 당신이 촘촘해요"(종합)") 라는 기사의 댓글에 "송중기만 생각남~~" 이라는 댓글을 보게 되면 편향성 공격성이 상관없는 none none의 라벨링이 예측되지만 실제 값은 편견에 대한 라벨링은 none 이 맞지만 공격성에는 offensive가 라벨링이 되어 있다. 이와 같은 데이터 셋을 특성으로 볼 때 높은 정확도를 얻기는 쉽지 않다. SVM이 결코 데이터 셋의 특징을 잘 뽑지 못하여 정확도 측면이 낮은 것은 아니라 판단한다. 단일 LSTM 경우는 이러한 데이터 특성을 더 잘 나타내지 못하여 정확도 측면에서 SVM 보다 성능이 떨어졌다. 한편, CNN+LSTM을 사용하여 정확도를 약 88% 나타내는 것을 볼 수 있다.

6. 2 결론

본 모델에서 댓글 분석에 LSTM과 CNN을 결

합하여 제안하였다. 그 결과, 제안 모델의 결과 값이 약 88%로 다른 모델이나 선행 방법보다 안정적이고 정확도가 뛰어났다. 기계학습법 중 SVM을 이용한 방법도 기존 딥러닝 방법 LSTM 보다도 성능이 우수하였지만, SVM은 시퀀스 모델이 아닌 벡터형으로 변형해야 하는 단점이 있다. 시퀀스 모델을 사용하게 되면 단어별 출현 횟수에 따라 만들어진 특징 벡터가 벡터의 차원이 지나치게 커질 수 있는 문제 즉, 차원의 저주에 빠질 수 있게 된다. 정보 손실이나 유실이 많이 될 수 있다. 이에 비해 CNN은 특징을 추출할 때, 있어 시퀀스 모델을 통해 네트워크 순서, 컨볼루션 커널 크기를 word embedding 에 따라 구축할 수 있다. 시퀀스를 이용할 때에는 차원의 저주를 극복할 수 있을 뿐만 아니라 원하는 문장을 자유롭게 분절 및 병렬로 나누어 corpus화 하여 훈련을 진행할 수 있다. 또한 시퀀스를 Avg Pooling을 통하여 관련성 있는 데이터를 평균화 하여 표현할 수 있다. 조합 모델에서는 LSTM을 함께 사용하는데 이는 3개의 게이트를 이용하여 입력과 출력 시기를 조절 할 수 있다. 특히 이를 이용함으로써 새로운 문장을 응용할 때, 다음 단어를 미리 예측할 수 있는 방식으로 학습할 수 있다. LSTM이 학습할 때, 확률분포 뿐만 아니라 말뭉치로부터 긍정과 부정으로 예측된 다음 단어를 미리 예측하여 댓글 분석을 진행할 수 있는 장점 때문에 비교모형보다 정교한 결과를 나타 낸 것으로 판단된다.