

심층 신경망 기반 자연어처리 모델의 성능 비교

이 태 겸*, 신경 섭^o

Performance Comparison of Natural Language Processing Model Based on Deep Neural Networks

Taegyom Lee*, Kyungseop Shin^o

요 약

기존 자연어처리 분야의 다른 구조를 갖는 심층 신경망들은 같은 말뭉치를 학습했음에도 말뭉치의 특성에 따라 다른 성능을 보여주었다. 말뭉치를 학습했을 때 일반적으로 준수한 성능을 내는 모델을 찾기 위하여, 본 논문에서는 구성이 다른 자연어 말뭉치를 심층신경망에 학습시켜 말뭉치별 심층신경망의 정확도를 다양한 측면에서 비교한다. 실험에 사용한 심층신경망은 CNN과 RNN 그리고 CNN과 RNN을 합친 모델까지 총 3가지 신경망을 사용했다. 실험에서 사용한 말뭉치는 총 3개이며 말뭉치는 각기 다른 구성을 하고 있다. 말뭉치의 구성언어에 따른 모델의 정확도 비교실험과 학습말뭉치 크기에 따른 모델의 정확도 비교실험, 단어 임베딩 방법에 따른 모델의 정확도 비교 실험을 진행하여 구성이 다른 자연어 말뭉치에 효율적인 심층신경망 선택을 위한 지침을 제공한다.

Key Words : Natural Language Processing, Convolution Neural Networks, Recurrent Neural Networks, Bidirectional-RNN, Word2vec.

ABSTRACT

Deep neural networks with different structures in natural language processing showed different performance results even though they utilize training set with the same corpus. For this reason, before improving the structure of the model, researchers go through the process of finding a model that show good performance in general. In this paper, the accuracy of the deep neural networks is compared in various aspects with different corpus type. For our experiments, we used three neural networks and three corpus. The first model is CNN and the second model is RNN. Finally, the third model, CNN and RNN combined model. We compared the accuracy of models in terms of different data sizes, different word expressions, and different corpus, so that our work provides guidelines for selecting the appropriate neural network for natural language corpus via three comparison methods.

I. 서 론

최근에는 실생활에 유용한 인공지능 스피커, 텍스트 인식 업무처리 챗봇 등 자연어처리 문제에 딥러닝

방법을 적용한 기술들이 실생활에서 사용되고 있다. 딥러닝의 겨울을 지나 집중조명을 받기 시작한 후 현재까지 자연어처리 분야는 많은 연구가 진행되어왔다^[1]. 자연어는 일상생활에서 사용하는 언어를 말하며,

* 본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2018R1C1B50438 99).

※ 실험에 사용한 코드는 온라인(<https://github.com/TaeGYEOM/Comparison-of-Accuracy-of-Model>)에 공개함.

• First Author : Semyung University School of Computer Science, showm321@gmail.com, 학생회원

◦ Corresponding Author : Semyung University School of Computer Science, ksshin@semyung.ac.kr, 정회원
논문번호 : 201905-078-C-RU, Received May 13, 2019; Revised May 16, 2019; Accepted May 20, 2019

자연어처리는 인공지능 기술의 한 분야로 형태소 분석, 품사 태깅, 구문 분석 등 다양한 기술을 통해 자연어의 이해와 생성 및 분석을 다루는 커뮤니케이션 기술을 말한다. 이러한 자연어처리에 딥러닝 방법을 적용하여 심층신경망을 학습시킬 때 사용하는 데이터를 말뭉치라고 한다.

자연어처리를 위한 몇 가지 딥러닝 연구들은 심층 신경망의 구조를 변형시켜 성능향상을 보여주었다. CNN (Convolution Neural Networks) 구조를 이용한 감성 분석 연구^[2]는 주로 이미지 처리에 사용했던 깊은 층의 CNN^[3]을 얇은 층의 구조로 바꿔 사용했다. Convolution layer와 Pooling layer 각각 하나씩만 사용한 얇은 구조와 Word2Vec^[4]을 함께 사용해 성능향상을 보여주었다. Bi-RNN (Bidirectional Recurrent Neural Networks)^[5]은 순방향 RNN (Recurrent Neural Networks)과 역방향 RNN을 함께 사용한 구조이다. 기존 RNN에서 역방향 패턴까지 학습하여 하나의 RNN을 사용한 구조보다 성능을 높였다. RNN과 CNN 동시에 사용한 구조^[6]는 동일한 말뭉치를 학습했을 때 단일 구조로 사용한 RNN과 CNN의 성능보다 더 높았다. 신경망의 성능을 높인 연구들은 신경망의 구조를 개선하여 성능을 높였다. 또한, 동일한 말뭉치에서 신경망마다 다른 성능을 보였다. 따라서 동일한 말뭉치를 학습한 신경망들의 성능이 다르기 때문에 먼저 말뭉치에 적절한 모델 선정 과정이 필요하다. 본 논문에서는 말뭉치에 적절한 모델을 선정하기 위한 말뭉치별 각 심층신경망 성능 비교 정보를 제공한다. 자연어처리 연구에 사용된 서로 다른 구조의 두 가지 심층신경망과 두 가지 신경망을 합친 신경망까지 총 3개의 신경망을 사용한다. 신경망의 성능 비교를 위한 말뭉치는 두 개의 영화평가 감성 분석 말뭉치와 46개의 주제로 구성된 뉴스 토픽 말뭉치까지 총 3개의 서로 다른 구성을 가진 말뭉치를 학습한다. 학습에 사용되지 않은 별도의 말뭉치로 모델의 성능을 평가하며 평가지표로서 분류 정확도를 사용한다. 말뭉치 구성에 적절한 모델 선정을 위해 여러 방법으로 모델의 정확도 비교 실험을 진행했다. 말뭉치의 구성언어에 따른 모델의 성능 비교를 위해 언어가 영어 또는 한글인 영화감성분석 말뭉치 두 개를 3가지 모델에 학습시켜 정확도를 비교하였다. 말뭉치 표현에 따른 모델의 성능비교를 위해, 말뭉치에 Word2Vec 방식을 적용해 학습시킨 모델의 정확도와 적용하지 않고 학습시킨 모델의 정확도를 비교했다. 이와 더불어, 단어 임베딩 값을 모델 학습 시 재학습시켰을 때의 정확도와 재학습하지 않았을 때의 정확도를 비교하였다. 또

한, 학습말뭉치 크기에 따라 성능변화를 보기 위해 학습말뭉치 속 문장의 개수를 5,000개부터 25,000개까지 5,000개씩 증가시켜 정확도 비교실험을 수행하였다.

II. 자연어처리 모델

2.1 CNN

CNN은 각기 다른 크기의 필터가 일정한 간격으로 움직이며 합성곱 연산을 수행하는 신경망이다. 필터는 연관시키는 단어의 개수에 따라 크기가 다르다. 필터의 크기가 3이면, 단어 3개 사이의 연관성을 학습하여 어떤 범주의 패턴을 학습할 수 있다. CNN의 입력값으로 사용한 말뭉치는 단어 임베딩으로 표현했다. 단어 임베딩이란 텍스트를 구성하는 단어들을 하나의 인덱스 정수 또는 실수 벡터로 나타내는 것을 말한다. 본 연구에서는 단어 임베딩을 적용한 각각의 단어를 벡터화하였다. 그림 1은 본 논문에서 사용한 CNN 구조를 도식화하였다.

i 번째 단어가 k -dimensional vector인 x_i 로 표현될 때, 입력 레이어는 문장의 길이 n 만큼 k -dimensional word vector (x_i)를 연결하여 표현한다. 따라서 길이 n 을 가진 문장을 식 (1)과 같이 표현할 수 있다.

$$x_{1, n} = x_1 \oplus x_2 \oplus \dots \oplus x_n \quad (1)$$

Convolution layer에서 합성곱 연산은 필터 크기 h 를 기준으로 연산 된다. f 는 각 layer에 들어온 입력을 다음 layer로 전달하는 활성화 함수의 일종인 ReLU 함수이며, 필터의 가중치 w 에 따른 결과 c_i 는 식 (2)와 같다. 하나의 문장을 모두 연산했을 때 식 (3)

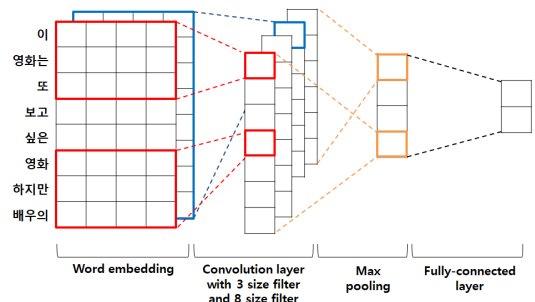


그림 1. 합성곱 신경망 모델
Fig. 1. Convolution neural network model

과 같은 Feature map c 가 만들어진다.

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad (2)$$

$$c = [c_1, c_2, \dots, c_{n-h+1}] \quad (3)$$

이 과정에서 단어 사이의 연관성이 생기고 필터의 개수만큼의 Feature map이 생성된다. Feature map에서 가장 큰 값들을 모아 Fully-connected layer에 넣어 긍정 또는 부정을 예측한다.

2.2 Bi-GRU

RNN은 현재 입력과 과거의 기억을 동시에 고려하는 신경망이다. 즉, 메모리의 개념이 추가된 신경망이라고 할 수 있다. 문제는 RNN의 메모리에 정보가 많아질수록 경사 소실 문제가 발생한다^[7]. 실험에서 사용하는 말뭉치 속 문장의 길이는 짧은 문장과 긴 문장이 섞여 있다. RNN의 Basic cell은 문장이 길어져 저장할 정보가 많아질수록 경사 소실 문제가 발생하여 모델의 정확도가 떨어진다. 해결 방법으로 과거의 기억에 중요도를 두어 앞으로 기억할 정보에 과거의 기억을 얼마나 반영할 것인지를 결정해 경사 소실 문제를 감쇠시킨 GRU(Gated Recurrent Units) cell^[8]이 있다. 식 (4)는 GRU cell의 정보 기억 방법이다.

$$\begin{aligned} z_t &= \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \\ r_t &= \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \\ \tilde{h} &= \tanh(r_t \circ U\tilde{h}_{t-1} + Wx_t) \\ h_t &= (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \end{aligned} \quad (4)$$

새로운 입력 x_t 와 과거의 기억 h_{t-1} 을 더해 새로운 기억 \tilde{h} 를 생성한다고 했을 때 h_{t-1} 의 반영 비율

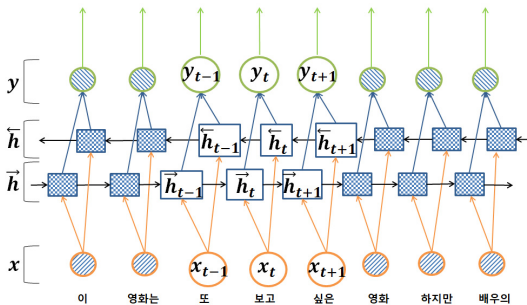


그림 2. 양방향 RNN 모델
Fig. 2. Bidirectional RNN model

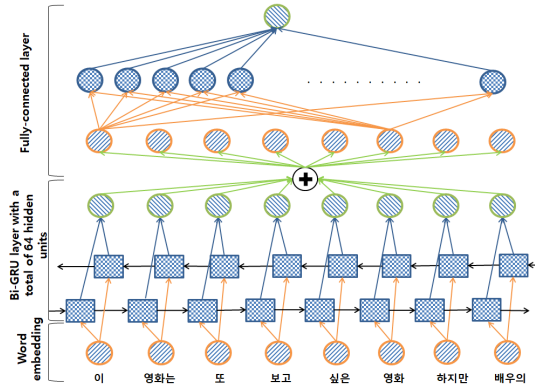


그림 3. 양방향 GRU 모델
Fig. 3. Bidirectional GRU model

을 조절해주는 Reset gate r_t 와 h_{t-1} 과 \tilde{h} 를 더해 현재의 기억 h_t 를 만들 때 h_{t-1} 과 \tilde{h} 의 비율을 정하는 Update gate z_t 까지 총 두 개의 Gate를 사용해 다음 상태 h_{t+1} 에서 기억할 정보와 삭제할 정보를 결정한다.

RNN은 과거의 정보를 저장해 과거에서부터 현재까지 순방향으로 입력된 데이터의 패턴을 학습할 수 있다. 만약 미래에서부터 현재의 흐름의 역방향 패턴까지 함께 고려할 수 있다면 더 높은 확률로 정답을 찾아낼 수 있다. Bidirectional RNN^[5]은 데이터를 입력받아 서로 반대되는 방향으로 처리하여 새로운 패턴을 학습한다. 그림 2는 Bi-RNN의 구조이다.

본 논문의 실험에서는 긴 문장들이 있기 때문에 경사 소실 문제를 극복한 GRU cell을 사용하고 형태소들이 문장을 구성하는 순서가 다른 영어와 한글의 차이를 보완하기 위해 Bidirectional GRU를 사용한다. 그림 3은 실험에서 사용한 Bi-GRU 구조이다. Bi-GRU 층의 출력값은 고차원의 표현이기 때문에 차원의 크기를 줄여 Fully-connected layer의 입력으로 사용한다.

2.3 Bi-GRU + CNN

CNN은 시퀀스 데이터를 입력으로 받아서, 보다 짧은 길이의 필터로 데이터의 패턴 정보를 압축 저장할 수 있다. RNN은 긴 길이의 데이터보다 짧은 길이의 데이터에서 정확도가 더 높다. 두 신경망의 장점을 극대화하기 위해 순차적으로 연결한 구조를 사용한다면 단일 구조의 신경망보다 좋은 결과를 낼 수 있다. 본 논문에서는 두 모델의 결합을 통한 서로 다른 말뭉치에서 범용성 확인을 하고자 Bi-GRU와 CNN을 순차

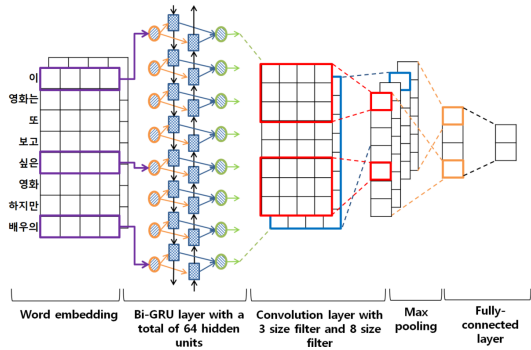


그림 4. 양방향 GRU와 합성곱 신경망을 합친 모델
Fig. 4. Bi-GRU and CNN ensemble model

적으로 연결한 구조를 사용하며 구조는 그림 4와 같다.

III. 실험 방법 및 말뭉치

3장에서는 실험 방법과 사용한 말뭉치에 대한 정보를 소개한다.

3.1 말뭉치

1) NSMC^[9] : 네이버 영화 리뷰 말뭉치로 150,000개의 문장으로 이루어져 있으며 실험에서는 15,000개의 문장을 분리해 135,000개의 학습말뭉치와 15,000개의 평가말뭉치로 구성하여 사용했다. 문장마다 긍정적인 문장이면 1, 부정적인 문장이면 0이 라벨링 되어 있다.

2) IMDB^[10] : 영어로 된 영화 리뷰 말뭉치로 25,000개의 학습말뭉치와 25,000개의 테스트말뭉치로 나뉘어 있다. 긍정적인 문장과 부정적인 문장이 분리되어 있으며, 각 문장별로 긍정은 1, 부정은 0으로 라벨링 하여 사용한다.

3) Reuters^[11] : 영문 뉴스 기사 말뭉치로 원본 로이터 말뭉치의 135개 토픽 중 샘플이 많은 순서대로 총 46개의 토픽을 선정해 만든 말뭉치를 사용했다. 1부터 46까지의 숫자로 문장별 라벨링이 되어있다. 각 범주에 데이터 개수가 일정하지 않아 더 높은 정확도를 위해서는 더욱더 세밀한 데이터 전처리가 필요하다. 본 논문의 실험에서는 있는 그대로의 말뭉치를 사용했다.

3.2 실험 방법

1) W2V : 인공지능이 자연어를 이해하기 위해서는 숫자로 바꿔 표현해야 한다. 말뭉치에 있는 모든 단어를 벡터로 표현해주는 방법으로 Word2Vec을 사용한

다. Word2Vec은 문장 속 특정 단어와 윈도우 크기만큼의 주변 단어들 사이 출현 확률을 학습하여 통계적 방법으로 단어를 벡터로 표현하는 방법이다. 벡터로 표현된 단어들을 좌표 차원에 나타내면 연관성이 있는 단어들은 비슷한 좌표를 가진다. 비슷한 좌표를 가진 단어들이 한 문장 속에 많으면 정밀도가 높은 벡터라고 할 수 있다. 실험에서는 정밀도가 높은 벡터를 더 잘 찾는 Skip-gram^[12] 방식을 사용했다.

2) Random : W2V 방법을 사용하지 않고 단어 벡터를 랜덤한 값으로 초기화하고 이 값을 모델의 학습 과정에서 함께 학습하는 방법을 사용한다. 모델이 학습할 때 함께 단어 벡터를 학습시키는 것을 재학습이라 하고 본 논문에서는 Update라고 표기했다.

3) W2V+Update : W2V 방법과 같이 Skip-gram 방식을 사용해 단어 벡터를 초기화하고 모델의 학습 과정에서 단어 벡터를 재학습하였다.

4) Data size : W2V 방법으로 초기화시킨 단어 벡터를 사용했다. 학습 과정에서 재학습은 하지 않았다. 학습말뭉치의 크기를 다르게 하여 실험을 진행했다. 학습말뭉치는 IMDB와 NSMC 말뭉치를 사용했고 긍정적인 문장과 부정적인 문장 비율이 반씩 되도록 했다. IMDB 말뭉치의 총 문장의 개수는 25,000개이므로 W2V에서 학습한 정확도와 같기 때문에 한 번 더 실험하지 않았다. Reuters는 46개 토픽 중에 문장이 5,000개를 포함한 토픽이 없기 때문에 실험을 진행하지 않았다.

이진 분류 문제에서 손실함수는 Binary cross-entropy, 최적화 함수는 Adam, 활성화 함수는 Sigmoid를 사용하였다. 다중 분류 문제에서 손실함수는 Categorical cross-entropy, 최적화 함수는 Rmsprop, 마지막 출력층의 활성화 함수는 Softmax를 사용했다.

IV. 실험 결과 및 분석

실험 방법에 따른 모델의 정확도를 얻은 결과를 표 1에 정리했다. Word2Vec, Random, Update 등 데이터 표현방법에 따른 모델의 정확도 실험과 문장 5,000개부터 25,000개까지 말뭉치 크기에 따른 모델의 정확도 실험을 진행했다. 그림 5는 두 가지 임베딩 방법에 따라 학습한 Bi-GRU와 CNN의 가장 높은 정확도를 말뭉치별로 비교한 그래프이다. CNN에 W2V 방법을 적용한 말뭉치의 정확도와 Random 방법을 적용한 말뭉치의 정확도를 뺀 값의 평균은 0.43%로 정확도의 변화가 크지 않다. 하지만 Bi-GRU의 정확도 차

표 1. 데이터 크기와 학습 방법이 다른 말뭉치들에 대한 3가지 모델의 정확도
Table 1. Comparison of accuracy of three models for corpus with different data size and learning method

		Representation			Data size				
		W2V	W2V +Update	Random +Update	5,000	10,000	15,000	20,000	25,000
IMDB	CNN	87.1	89.2	88.5	82.4	85.7	86.3	87.0	-
	Bi-GRU	83.7	87.9	86.8	76.0	77.5	78.5	80.7	-
	Bi-GRU+CNN	88.5	89.7	88.9	83.7	86.4	87.6	88.0	-
NSMC	CNN	77.0	81.6	82.5	58.4	59.6	58.8	58.2	58.8
	Bi-GRU	79.2	82.0	70.0	58.8	61.0	61.2	60.8	61.1
	Bi-GRU+CNN	79.6	82.9	82.8	59.6	64.6	66.7	67	68
Reuters	CNN	73.0	79.4	77.9	-				
	Bi-GRU	72.5	76.0	59.5					
	Bi-GRU+CNN	75.7	81.2	78.9					

이의 평균은 9.86%로 CNN보다 변화가 크다.

그림 6은 말뭉치 크기에 따른 두 종류의 말뭉치를 학습한 모델들의 정확도 변화를 보여준다. 학습하는 데이터의 크기가 클수록 정확도가 점점 높아진다. NSMC 말뭉치를 학습한 Bi-GRU+CNN의 정확도는 말뭉치의 크기 증가에 따라 평균 2.1%의 가장 높은 정확도 상승률을 보였다. 모든 말뭉치에서 가장 높은 성능은 두 신경망을 합친 구조이다.

그림 7은 학습말뭉치에 따른 모델별 정확도 비교 그래프이며, 세 번째 결과인 Ensemble은 Bi-GRU와 CNN을 합친 모델을 의미한다. 모든 말뭉치는 단어를

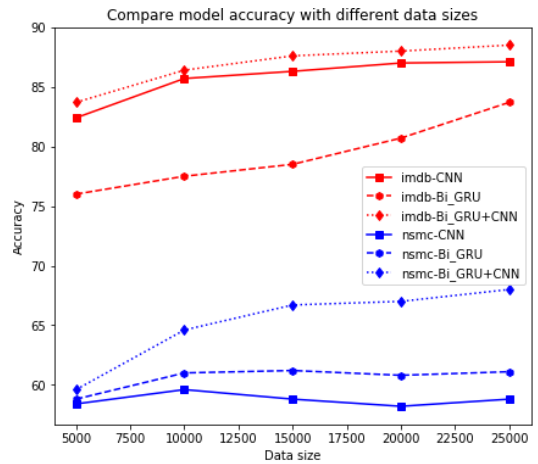


그림 6. 다른 말뭉치 크기에서 모델 정확도 비교
Fig. 6. Comparison of accuracy of model learning different data sizes

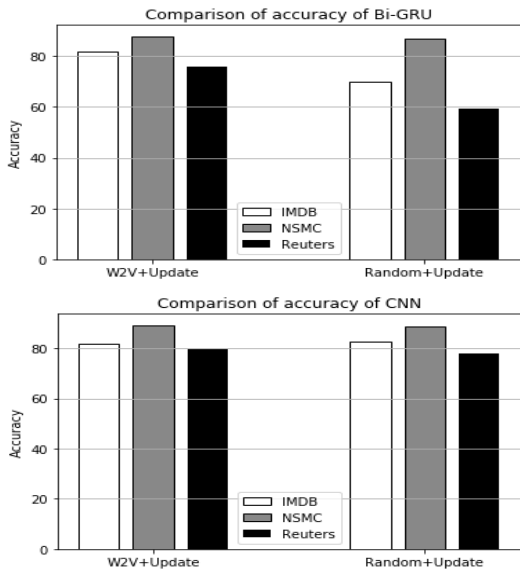


그림 5. 학습 방법에 따른 CNN과 Bi-GRU의 정확도 비교
Fig. 5. Comparison of accuracy of CNN and Bi-GRU

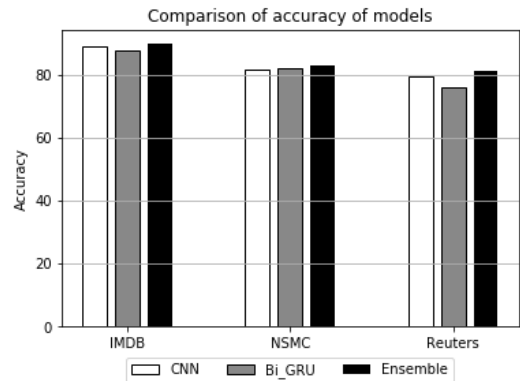


그림 7. 말뭉치 종류에 따른 모델 정확도 비교
Fig. 7. Comparison of accuracy of model learning different corpus type

명사, 동사 등 형태소별로 구분해 태그를 달아주는 형태소 분석을 하지 않고 오타만 제거하여 띄어쓰기를 기준으로 단어를 나누고 W2V 방법을 사용해 실험을 진행했다. 실험에서 사용한 말뭉치들의 크기는 NSMC 말뭉치가 135,000개의 문장으로 구성되어 있고 IMDB 말뭉치가 25,000개의 문장으로 구성되어 있으며 Reuters 말뭉치는 약 9000개 정도의 문장으로 구성되어 있다. 그림 6에서 말뭉치 속 문장의 개수가 많을수록 정확도가 증가했다. 하지만 IMDB 말뭉치를 학습시킨 모델들의 평균 정확도가 88.93%로 가장 높았고 NSMC와 Reuters 말뭉치를 학습한 모델들의 평균 정확도는 각각 82.16%, 78.86%로 비교적 낮았다. IMDB와 Reuters는 영어로 구성된 말뭉치이고 NSMC는 한국어 말뭉치이다. 그림 7을 보면 CNN은 영어 말뭉치에서 정확도가 Bi-GRU보다 각각 1.3%, 3.4% 높았고, Bi-GRU는 한국어 말뭉치인 NSMC에서 CNN보다 0.4% 높았다.

V. 결론

본 논문에서는 말뭉치를 표현하는 방법, 말뭉치의 크기, 말뭉치의 구성언어에 따른 심층신경망들의 정확도를 비교했다. 말뭉치에 Word2Vec을 적용했을 때 성능이 향상되는 구조가 있었지만 CNN 모델 일부에서는 Word2Vec을 사용한 방법이 오히려 맞지 않는 구조도 있었다. 말뭉치 크기에 따라 성능이 점점 좋아지는 것을 그래프를 통해 확인했다. 한글 말뭉치에서는 Bi-GRU가 CNN보다 좋은 성능을 낸다는 것을 확인했고, 영어 말뭉치에서는 CNN이 Bi-GRU보다 더 나은 성능을 보여주었다. 서로 다른 말뭉치에서 강점을 보이는 두 개의 신경망을 결합한 구조는 시간이 오래 걸린다는 단점이 있지만 모든 실험에서 정확도가 가장 높음을 보였고 결합 구조는 범용성을 갖게 해주는 것을 실험적으로 확인했다. 본 논문의 결과를 통해 말뭉치의 구성에 따른 학습 방법의 선정과 적합한 모델을 선택하는 데에 도움을 줄 수 있을 것이다.

References

[1] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intell. Mag.*, vol. 13, no. 3, pp. 55-75, Jul. 2018.

[2] Y. Kim, "Convolutional neural networks for

sentence classification," in *Proc. 2014 Conf. EMNLP*, pp. 1746-1751, Oct. 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Inf. Process. Syst.*, pp. 1097-1105, Dec. 2012.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR Workshop 2013*, May 2013.

[5] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Sign. Process.*, vol. 45, no. 11, pp. 2673-2681, Nov. 1997.

[6] X. Wang, W. Jiang, and Z. Luo, "Combination of convolutional and recurrent neural network for sentiment analysis of short texts," in *Proc. 26th Int. Conf. Computat. Linguistics*, pp. 2428-2437, Dec. 2016.

[7] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107-116, Apr. 1998.

[8] J. Y. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555, 2014.

[9] Lucy Park, *Naver sentiment movie corpus v1.0(2016)*, <https://github.com/e9t/nsmc>.

[10] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting of the Assoc. for Computational Linguistics*, pp. 142-150, Jun. 2011.

[11] D. D. Lewis, *Reuters-21578 Text Categorization Collection Data Set(1997)*, <https://bit.ly/2JPwSa0>.

[12] I. H. Kim and B. C. Jang, "Effective word embedding for twitter data," *J. KICS*, vol. 43, no. 11, pp. 1903-1910, Nov. 2018.

이 태 겹 (Taegygeom Lee)



2018년 2월 : 세명대학교 컴퓨
터학부 졸업

2018년 2월~현재 : 세명대학교
컴퓨터학과 석사과정
<관심분야> 자연어처리, 인공
지능, 데이터사이언스

[ORCID:0000-0002-2384-1916]

신 경 섭 (Kyungseop Shin)



2011년 1월 : KAIST 전기 및
전자공학과 석사

2015년 1월 : KAIST 전기 및
전자공학과 박사

2015년 2월~2017년 8월 : KT
융합기술원 인프라연구소
5G TF 선임연구원

2017년 9월~현재 : 세명대학교 컴퓨터학부 조교수
<관심분야> 차세대 이동통신, IoT 네트워크, 무선전
력전송

[ORCID:0000-0002-3867-1921]