가짜 뉴스 기사 분류

김재형 O, 황정현, 최지연, 서예진, 이현지
세종대학교 스마트기기공학과
hiddencraft@naver.com, rjhwang08@gmail.com
choiwldus@naver.com, seoyj0325@naver.com, lily1445@gmail.com

요 약

정보화 시대에서 인터넷 상에는 기사 내용과는 다른 자극적인 헤드라인으로 사람들의 클릭을 유도하여 조회수를 올리는 가짜 뉴스가 많아지면서 여러 사회적 문제를 일으키고 있다. 본 논문에서는 인공지능 기술을 적용하여 이에 대한 해결책을 제시하고자 한다. 이를 위해 먼저 한국어의 특성을 고려하여 이에 적합한 기존 자료를 참고하였고, 이를 토대로 텍스트를 벡터로 표현해주는 워드임베딩 모델 중, 사전 훈련된 FastText 모델을 사용하였다. 또한 벡터화 된 데이터를 통해 가짜 뉴스를 분류할 모델로 CNN, LSTM 그리고 APS 이렇게 3 가지를 사용하여 학습시키고 검증하였다.

1. 서론

정보화 시대에서 인터넷 상에는 많은 양의 기사가 생산되고 있다. 하지만 기사 내용과는 다른 자극적인 헤드라인으로 사람들의 클릭을 유도하여 조회수를 올리는 '가짜 뉴스'(혹은 낚시성 뉴스)가 많아지면서 여러 사회적 문제를 일으키고 있다. 대표적 사례로 16 년 미국 45 대 대선을 들 수 있다. 미 대선 과정에서 '교황이 도널드 트럼프(Donald Trump)를 지지한다', '힐러리 클린턴(Hillary Clinton)이 테러단체 이슬람국가에 무기를 판매했다' 등의 가짜 뉴스가 소셜 미디어 등을 통해 집중적으로 유통되면서 선거 결과에 영향을 미친 것으로 분석되고 있다. 이렇듯 가짜 뉴스는 단지 개인에게 피해를 끼치는 수준을 넘어, 전세계적으로 간과할 수 없는 사회적 문제를 일으키고 있다. 따라서 본 연구에서는 인공지능 기술을 이용하여 가짜 뉴스의 분류를 통해 유용한 정보만 골라 소비할 수 있는 환경을 조성하고자 하였다

현재 전세계적으로 인공지능을 통한 가짜 뉴스 분류에 대한 연구가 활발하다. 하지만 해외의 모델을 그대로 적용하기에는 한국어는 영어에 비해 형태소 분석이 어렵고, 또한 같은 문장임에도 한국어 문장의 단어수가 영어 문장의 단어 수보다 적어 깊은 신경망을 운용할 수 있는 특징수가 부족할 수 있다.

따라서 본 연구에서는 좀 더 한국어에 잘 맞는 모델을 적용하기 위해 한국정보처리학회에서 주관한 2018 춘계학술발표대회 출품작인 '딥러닝 기법을 이용한 가짜뉴스 탐지' 팀의 모델을 참고하여, 워드 임베딩 모델인 'FastText'와 합성곱신경망(CNN) 이용하여 분류기를 제작하였다.

2. 관련 연구

2.1 Word Embedding

신경망을 비롯한 대부분의 머신러닝 알고리즘에서 텍스트를 적용하기 위해서는, 텍스트를 일련의 벡터 형태로 표현해야 한다. 이와 같이 텍스트를 의미 있게 표현하는 방법을 워드 임베딩이라고 하는데, 워드 임베딩(Word Embedding)을 위한 연구로는 Word2Vec, Glove, FastText 등이 있다.

일반적으로 영어는 공백을 기준으로 구분되는 단어를 임베딩 단위로 삼는다. 하지만 한국어는 어형 변화가 많기 때문에 형태소 단위로 워드임베딩을 수행했을 때 더 나은 성능을 보인다. 따라서 본 연구에서는 단어 단위로만 분석할 수 있는 Word2Vec 방식이 아닌 형태소 단위로 분석할 수 있는 FastText 방식으로 진행하였다.

그림1. FastText 의 단어 표현(bag of character)예

FastText에서 각 단어는 글자들의 n-gram으로 나타낸다. n의 숫자에 따라 단어들이 얼마나 분리되는지 결정된다. 예를 들어 n을 3으로 잡은 경우, where은 whe, her, ere로 분리하고 이들 또한 임베딩을 한다. 더 정확히 나타내기 위해 시작과 끝을 나타내는 <, >을 사용해 <wh,whe,her,ere,re> 5개와 <where> 하나를 추가적으로 더 임베딩한다.

FastText를 사용할 때는 n의 최소값과 최대값을 설정할 수 있고, FastText의 인공 신경망을 학습한 후에는 데이터 셋의 모든 단어의 각 n-gram에 대해서 워드 임베딩이 된다. 장점은 데이터 셋만 충분한다면 위와 같은 내부 단어를 통해 모르는 단어에 대해서도 다른 단어와의 유사도를 계산할 수 있다는 점이 모르는 단어에 대처할 수 없었던 Word2Vec와는 다르다고 할 수 있다.

FastText 방식을 사용하기 위해 먼저 데이터를 Tokenizing 할 필요가 있다. Tokenize 란 어미, 조사 등이 붙어 구분이 불가능한 한국어를 명사, 동사, 어미, 조사 등으로 형태소를 분석해 Word Embedding 에 적합한 데이터를 만드는 과정을 뜻한다. KoNlP 한국어 처리 패키지에는 다양한 형태소 분석기들(Hannanum, Kkma, Komoran, Mecab, Twitter)이 있으며, 각 분석기는 분석기마다 가지고 있는 Dictionary 를 기반으로 단어를 Tokenizing 한다. 본 연구에서는 이 중 Komoran 분석기를 활용했다.

표 1. 형태소 라벨 분류

분류	형태소	분류	형태소
NNG	일반명사	VCN	부정 지정사
NNP	고유명사	XSV	동사 파생 접미사
NNB	의존명사	XSA	형용사 파생 접미사
NR	수사	SN	숫자
VV	대명사	MAG	일반 부사
VA	형용사	MM	관형사
VCP	긍정지정사	MAJ	접속 부사

2.2 합성곱신경망(Convolutional Neural Network)

2.2.1 합성곱 신경망(Convolutional Neural Network; CNN)

합성곱 신경망은 뇌의 시각피질이 이미지를 처리하고 인식하는 원리를 차용한 신경망이다. 합성곱 신경망의 구조는 그림 2 와 같다.

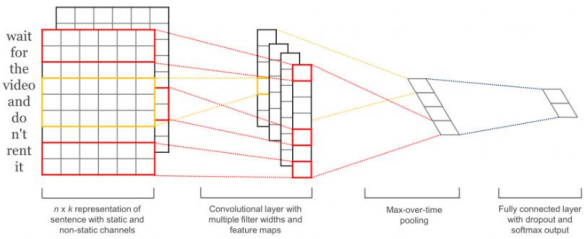

그림 2. CNN 의 구조

첫번째 층(layer)은 단어를 저차원으로 임베드한다. 그 다음 층은 여러 사이즈 필터를 이용해, 임베드된 단어 벡터에 대해 합성곱 변환을 수행한다. 그 다음 합성곱 층의 결과를 긴 피쳐 벡터로 Max pooling 하고, 완전연결계층(fully connected layer)에서 드롭아웃(dropout) 정규화와 softmax 를 통해 분류를 수행한다.

convolution 층은 입력 이미지에 대해 필터를 적용하여 필터링을 수행한다.

풀링 층(Pooling layer)은 컨볼루션 층의 출력 데이터를 입력으로 받아서 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용된다. 풀링 층을 처리하는 방법으로는 Max Pooling 과 Average Pooling, Min Pooling 이 있다. 정사각행렬의 특정 영역 안에 값의 최댓값을 모으거나 특정 영역의 평균을 구하는 방식으로 동작한다. CNN 에서는 주로 Max Pooling 을 사용한다.

마지막으로 fully connected layer(affine layer)를 생성한 후, 역전파를 이용해서 입출력간 오차를 최소화하는 방향으로 학습을 반복해 나간다.

본래 CNN 은 이미지 처리를 위해 만들어진 아키텍처이나, 이를 텍스트에 적용하는 연구도 활발하게 이루어지고 있다.

2.2.2 드롭아웃(Dropout)

드롭아웃(dropout)은 합성곱 신경망의 오버피팅(overfitting)을 방지하는 가장 유명한 방법이다. 드롭아웃 층은 뉴런의 일부를 확률적으로 '비활성화'한다. 이는 뉴런의 상호 적응을 방지하고 피쳐를 개별적으로 학습하도록 강제한다. 드롭아웃을 학습 중에는 0.5, 평가 중에는 1 로 비활성화 한다.

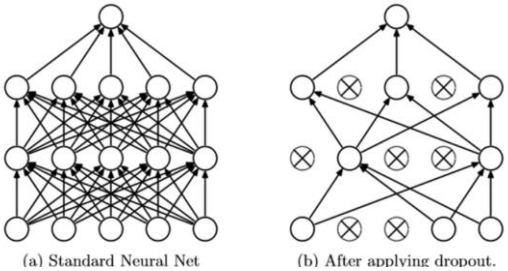

그림 3. 일반적 신경망 구조(a)와 드롭아웃 (Dropout)이 적용된 신경망의 구조(b)

2.3 순환신경망(Recurrent Neural Network)과 LSTM

2.3.1 순환신경망(Recurrent Neural Network; RNN)

순환신경망(RNN)은 음성 인식, 필기체 인식, 제스처 인식과 같이 시간에 따라 연속성을 가지는 데이터 혹은 길이가 가변인 데이터를 학습 및 표현하는 작업에 적합한 인공 신경망이다. 시퀀스 길이에 관계없이 입력과 출력을 받아들일 수 있는 네트워크 구조이기 때문에 필요에 따라 다양하고 유연하게 구조를 만들 수 있다는 점이 RNN 의 가장 큰 장점이다.

RNN 은 은닉층(Hidden Layer)의 결과가 다시 같은 은닉층의 입력으로 들어가도록 연결되어 있는 구조를 가진다. 이러한 구조는 인공신경망으로 하여 은닉층에 저장되어 있는 과거의 정보들을 현재의 입력 값과 결합하여 사용할 수 있게 함으로써 입력 값 시퀀스에 대해 시퀀스 내부 유닛 간의 상호적인 정보를 어느 정도 유지시킨 은닉변수(Hidden state) 시퀀스로 비선형 변환을 시킬 수 있다.

그림 4 에서는 일반적인 RNN 의 구조를 나타내고 있다.

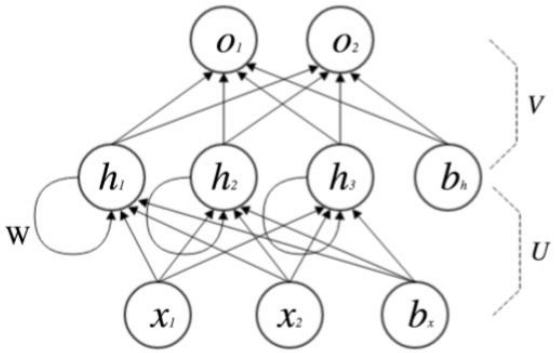

그림 4. RNN 의 구조

2.3.2 장단기 기억 모델 (Long Short-Term Memory models)

전통적인 순환신경망(RNN)은 관련 정보와 그 정보를 사용하는 지점 사이 거리가 멀 경우 역전파시 기울기(gradient)가 점차 줄어 학습능력이 크게 저하되는 것으로 알려져 있는데 이를 기울기 값이 사라지는 문제(vanishing gradient problem)라고 한다. 이 문제를 극복하기 위해서 고안된 것이 바로 장단기 기억(Long Short-Term Memory models; LSTM)이다.

LSTM 은 전통적인 RNN 과 마찬 가지로 은닉변수(Hidden state)를 거쳐 최종 출력 값을 계산하지만, 은닉변수의 계산 과정에서 입력게이트(Input Gate), 잊기게이트(Forget Gate), 출력게이트(Output Gate) 3 종류의 게이트 유닛들을 적절하게 이용해서 정보의 흐름을 조절하는 형태를 가진다.

그림 5 에서는 LSTM 의 구조를 나타내고 있다.

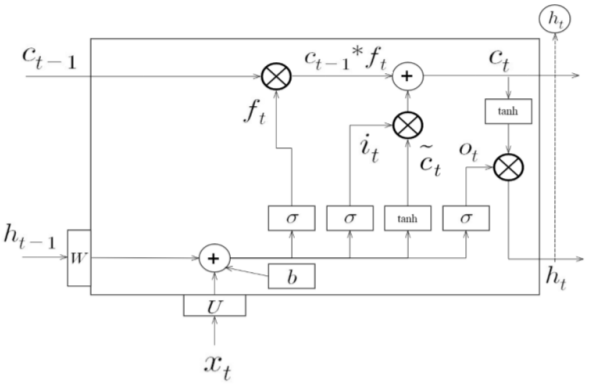

그림 5. LSTM 의 블록(Block) 구조

식(1)에서 볼 수 있듯 잊기게이트는 h_{t-1}, x_t 와 바이어스(Bias)의 가중합(Weighted Sum)에 Sigmoid 를 씌운 형태이며, 소자변수(Cell State)가 가지고 있는 각각의 정보에 대해 0 과 1 사이의 값을 계산한다. 여기서 1 은 해당 위치의 정보를 전부 다 기억한다는 의미를 가지며, 0 은 반대로 모두 잊어버리겠다는 의미를 가진다.

입력 게이트는 얼마만큼의 새로운 정보를 소자변수로 가져갈지 결정하는 값이다. 새로운 정보도 마찬가지로 h_{t-1}, x_t 와 바이어스의 가중합을 통해 계산되며 tanh 활성화 함수를 통해 (− 1, + 1) 사이의 값으로 전환된다. 식(2)와 식(3)을 통해 알 수 있다. 마지막으로 출력게이트는 필터링(Filtering)을 통해 계산된 소자변수의 유용한 정보를 최종 은닉변수로 출력하게 된다.

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

$$i_t=\sigma(U^i x_t + W^i h_{t-1} + b^i) \quad (2)$$

$$\tilde{c_t} = \tanh(U^c x_t + W^c h_{t-1} + b^c) \quad (3)$$

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t \quad (4)$$

$$o_t = \sigma(U^o x_t + W^o h_{t-1} + b^o) \quad (5)$$

$$h_t = o_t * \tanh(c_t) \quad (6)$$

LSTM 의 경우, 학습 능력을 크게 향상시켜 연속적이고 길이가 긴 학습 데이터가 가지고 있는 순차적 의존성을 효과적으로 학습할 뿐만 아니라 과거 시간 단계 정보의 중요도를 판별하여 얼마나 많은

과거 정보를 현재까지 끌어올 것인지, 혹은 현재 시간 단계의 정보를 얼마나 사용할 것인지 결정할 수 있기 때문에 긴 시간 동안 나타나는 입력 데이터의 의존성을 기존의 RNN 보다 더욱 효과적으로 학습하는 것이 가능하다.

본 연구는 양방향 장단기(Bi-directional LSTM; Bi-LSTM)모델을 활용할 것이다. Bi-LSTM 은 순차적 데이터 활용에 있어서 가장 많이 쓰이는 딥러닝 모형인 LSTM 을 두 개를 함께 학습시켜, 각 데이터에 대해 왼쪽(backward)뿐만 아니라 오른쪽(forward) 데이터를 고려하도록 보완한 모델이다.

2.4 Attentive Pooling Networks

2.4.1 Attentive Pooling

최근에 어텐션 메커니즘이 이미지 캡셔닝과 기계 번역에 성공적으로 사용되고 있다. 그러나 Pair-wise Ranking 이나 텍스트 분류와 같은 두 개의 입력을 가지는 자연어 처리(NLP) 작업에 어텐션 메커니즘을 적용한 더 나아간 연구는 없다. 하지만 어텐티브 풀링(AP)은 이러한 작업의 향상된 성능에 두 개의 입력의 유사성을 효과적으로 나타냄으로써 기여하고 있다. 문서 안에서 단어의 빈도를 통해 통계적으로 유사성을 측정하는 Term Frequency-Inverse Document Frequency(TF-IDF) 방법이 있지만 이 모델은 두 입력에 대해 같거나 비슷한 의미를 가진 단어의 가중치를 증가시키면서 유사성을 측정하는 차이가 있다.

2.4.2 Attentive Pooling Networks

어텐티브 풀링은 h 로부터 얻은 정보가 b 의 계산과 b 로부터 얻은 정보가 h 의 계산에 직접적인 영향을 주어 현재의 입력쌍(h,b)을 풀링 레이어가 알 수 있게 하는 방법이다.

AP 의 주요 내용은 입력쌍의 예측된 세그먼트의 유사도 측정값을 학습하는 것과 세그먼트 사이의 유사도 점수를 사용하여 어텐션 벡터를 계산하는 것이다.

그림 6 은 r^h와 r^b를 구성하기 위해 컨볼루션의 출력값에 AP 를 적용한 것이다. 제목의 크기가 M, 본문의 크기가 L 이고 입력쌍이 (h,b)일 때를 고려해 보자. 행렬 H 와 B 를 컨볼루션으로 계산한 후 다음과 같이 행렬 G 를 계산할 수 있다.

$$\mathrm{G} = \tanh(H^T \mathrm{UB}) \tag{7}$$

이 때 U 는 신경망에 의해 학습된 파라미터의 c*c 행렬이다. 컨볼루션이 H 와 B 를 계산하기 위해 사용되면 행렬 G 는 b 와 h 의 convolved k-size 컨텍스트 창 사이의 soft alignment 의 점수를 포함한다.

그 후에 벡터 g^h, g^b를 생성하기 위해 G에 행방향 풀링과 열방향 풀링을 각각 적용한다. 의례적으로 g^h와 g^b 벡터의 j번째 원소들은 다음 식과 같다.

$$[g^h]_j = \max_{1<m<M} [G_{j,m}] \tag{8}$$

$$[g^b]_j = \max_{1<l<L} [G_{l,j}] \tag{9}$$

각각의 벡터 g^b의 j원소는 제목 h에 관하여 본문 b의 j번째 단어 주변의 중요도 점수로 해석할 수 있다. 마찬가지로 벡터 g^h의 각각의 j원소는 본문 b에 관하여 제목 h의 j번째 단어 주변의 중요도 점수로 해석할 수 있다.

다음으로 벡터 g^h, g^b에 소프트맥스 함수를 적용하여 어텐션 벡터 σ^h, σ^b을 생성한다. 예를 들어 벡터 σ^h 의 j번째 원소는 다음과 같이 계산된다.

$$[\sigma^h]_j = \frac{e^{[g^h]_j}}{\sum_{1<l<M} e^{[g^h]_l}} \tag{10}$$

마지막으로 r^h와 r^b는 어텐션 벡터 σ^h, σ^b와 컨볼루션의 출력값 h, b를 각각 내적한 것과 같다:

$$r^h = H\sigma^h \tag{11}$$

$$r^b = \mathrm{B}\sigma^b \tag{12}$$

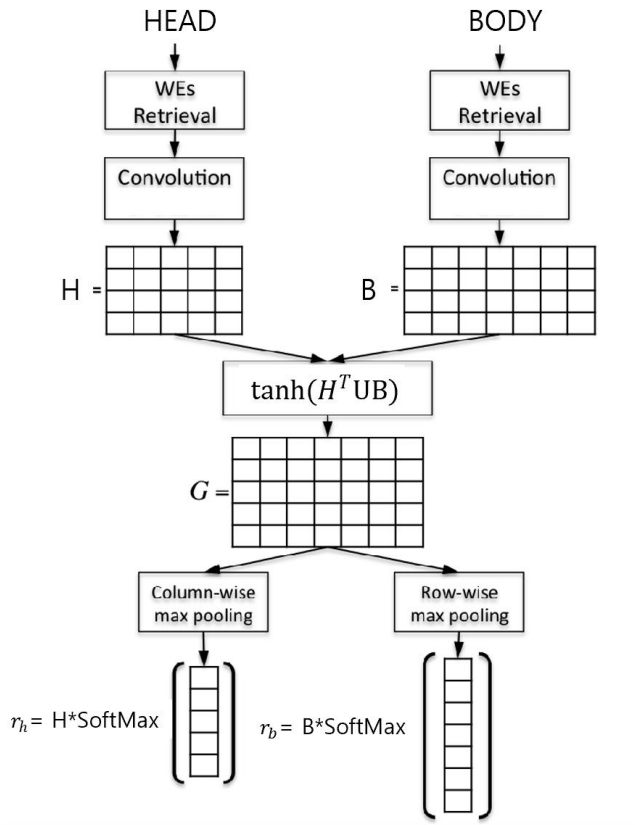

그림 6. Attentive Pooing Networks

3. 실험 결과 및 분석

3.1 데이터셋

본 연구에서는 앞서 언급한 '딥러닝 기법을 이용한 가짜뉴스 탐지' 팀에서 공유한 데이터셋을 사용하였다. 데이터셋은 중앙일보, 조선일보, 한겨레, 매일경제에서 주제를 두 가지로 구분하여 각각 31000 여 개, 68000 여 개의 뉴스기사를 크롤링하여 구성되었다. 본 연구에서는 이 중 31000 여 개의 '기사 제목과 본문이 부정합한 경우'에 대한 데이터셋을 이용하였다. 데이터는 언론사별로 경제, 사회, 정치, 연애, 스포츠로 카테고리를 나누어 동일 비율로 수집되었다. 진짜 기사와 직접 가공한 가짜뉴스 비율은 1:1 로 구성되어 있다. 그리고 학습데이터와 검증데이터는 8:2 비율로 구성되어 있으며, 신경망의 최종 정확도는 학습과 검증 데이터에 포함되지 않은 최신 기사 100 개를 진짜, 가짜 비율 1:1 로 가공하여 얻었다.

3.2 신경망 구성

본 연구에서 설계한 합성곱 신경망은 그림 2 와 같이 단어 임베딩으로 'Fasttext'에 의해 미리 학습된 벡터들을 사용한 합성곱 신경망이다. 제목과 본문에서 CONV (Convolution) layer 을 통해 특징 맵을 추출한 후, 추출된 특징 맵들을 POOL(Pooling)을 통해 하나로 만든다. 이후 FC(Fully Connected) layer 를 거쳐 분류를 진행한다.

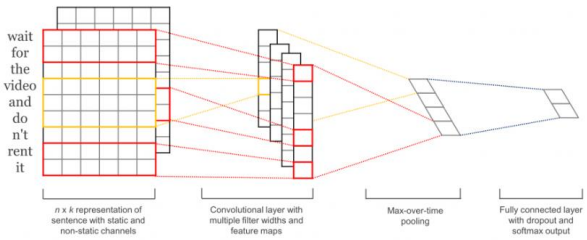

그림 2. CNN 의 구조

본 연구에서 설계한 양방향 장단기-합성곱 신경망(Bi-LSTM+BCNN)은 그림.7 과 같이 단어 임 베딩으로 'Fasttext'에 의해 미리 학습된 벡터들을 사용한 양방향 장단기-합성곱 신경망이다. 제목과 본문에서 LSTM 을 적용하고 CONV layer 을 통해 특징 맵을 추출한 후, 추출된 특징 맵들을 POOL(Pooling)을 통해 하나로 만든다. 이후 FC(Fully Connected) layer 를 거쳐 분류를 진행한다.

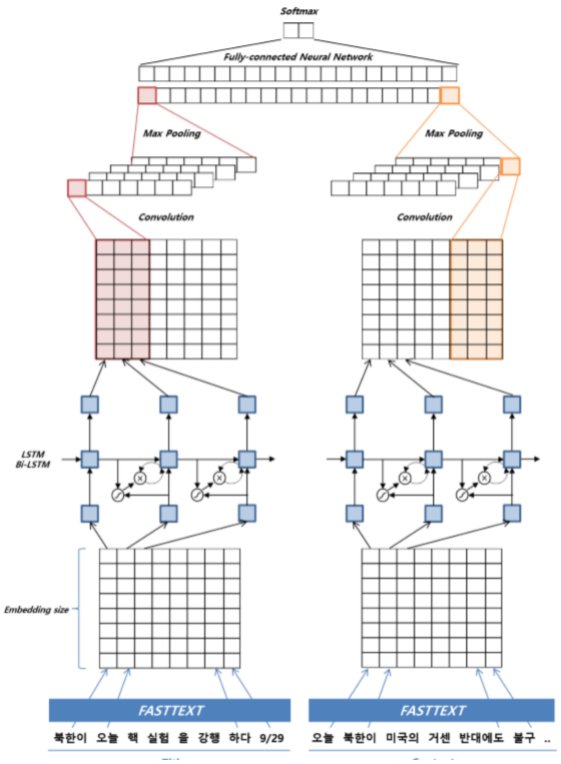

그림 7. Bi-LSTM + BCNN 의 구조

본 연구에서 설계한 BCNN with Attentive pooling similarity(APS-BCNN)은 그림 8 과 같이 BCNN 의 맥스 풀링에서 얻은 일차원 벡터 행렬에 어텐티브 풀링을 사용하여 표현한 두 개의 입력 사이에 유사성 벡터를 추가한 것이다. 유사성 벡터를 추가했기 때문에 향상된 성능을 기대해 볼 수 있다.

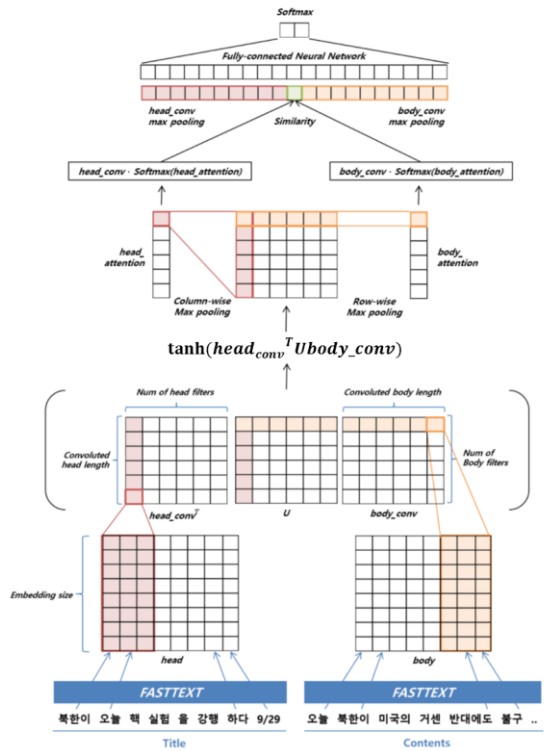

그림 8. Attentive Pooing Similarity

하이퍼파라미터는 표 2 와 같이 설정하였다. 설계 시 고려사항으로는 기사 제목과 본 문 내용 사이 텍스트 수 차이가 큰 것을 고려하여 필터 개수를 이에 비례하게 설정하였다.

표 2. 하이퍼파라미터

Lable	*Description*	*Optimized*
filter_size	필터의 크기	3
num_filters	필터 개수	256(제목) 1024(본문)
dropout	드롭아웃	0.5
L2_alpha	학습률	0.1
epoch	전체 데이터 실행수	20
batch_size	학습 미니배치 크기	128
embedding_Dim	단어 임베딩 벡터 차원 수	128

3.3 결과

각 모델에서의 평가 정확도는 표 3 과 같다.

표 3. 모델 정확도 비교

Model	*Accuracy*
BCNN	0.47
APS + BCNN	0.49
LSTM + BCNN	0.41

훈련에서의 정확도는 0.9 정도까지 달했으나 검증에서의 정화도는 0.6~0.7 평가에서의 정확도는 0.4 정도로 낮게 도출되었다. 평가 정확도가 학습 정확도보다 상당히 낮기 때문에 오버피팅(Overfitting) 하는 것으로 추측할 수 있다. 본 연구에서 사용한 훈련 데이터의 양이 적었기 때문에 발생한 것으로 보인다. 이를 해결하기 위해 dropout 을 추가하였으나 큰 변화를 보이지 않았다. 또한, LSTM 과 BCNN 을 결합한 모델에서는 가장 낮은 정확도를 보였다. 우리는 그 이유에 대해 다음과 같이 추측해 보았다. 기사의 제목은 ‘This apple is so tasty.’이고 본문에 ‘The red apple on the desk seems so tasty.’가 포함되어 있을 때, LSTM 을 적용한 후에는 각 문장에서 ‘apple’주위의 다른 단어들로 인해 이를 다른 벡터로 인식하여 정확도가 떨어진 것이다.

4. 결론

향후 모델 개선을 위해 사용할 수 있는 방법은 다음과 같다. 더 많은 데이터로 학습시키거나, 더 강력한 정규화 또는 더 적은 모델 파라미터를 사용하거나 앙상블 학습을 추가하는 것이다.

앙상블 학습(ensemble learning)은 머신러닝에서 여러 개의 모델을 학습시켜 그 모델들의 예측결과들을 이용해 하나의 모델보다 더 나은 값을 예측하는 방법을 말하며 앙상블 학습의 예로는 랜덤 포레스트, 배깅, 부스팅, 스태킹 등이 있다.

감사의 글

본 연구는 세종대학교 지능기전공학부 2019 년 1 학기 인공지능 수업을 통해 진행된 연구로 한 학기 동안 가르쳐 주신 최유경 교수님께 감사를 전합니다.

참고문헌

[1] 한국언론진흥재단 미디어연구센터, Media Issue 2017 년(3 권 3 호)

[2] 이동호, 이정훈, 김유리, 김형준, 박승면, 양유준, 신웅비, “딥러닝 기법을 이용한 가짜뉴스 탐지”, 한국정보처리학회 2018 춘계학술발표대회

[3] 최순영, Andrew Stuart Matteson, 임희석, “한국어-영어 법률 말뭉치를 이용한 로컬 이중 언어 임베딩”, 한국융합학회논문지, Vol9.No.10, pp.45-53, 2018

[4] 조형배, 서찬양, 김소영, 윤은영, “FastText 성능 향상을 위한 품사 정보 임베딩 기법 설계 및 구현”, 한국컴퓨터종합학술대회 논문집, Vol.2018 No.6[2018],pp.978-981, 2018

[5] 고동우, 양정진, “KoNLPy 와 Word2Vec 을 활용한 한국어 자연어 처리 및 분석”, 한국정보과학회 학술발표논문집, Vol.2018 No.6[2018], pp.2140-1142, 2018

[6] 김양훈, 황용근, 강태관, 정교민 “LSTM 언어모델 기반 한국어 문장 생성”, The Journal of Korean Institute of Communications and Information Sciences , Vol.41 No.05 , pp. 1-4, 2016

[7] 남석현, 함영균, 최기선 “한국어 특질을 고려한 단어 벡터의 Bi-LSTM 기반 개체명 모델 적용”, 제 29 회 한글 및 한국어 정보처리 학술대회 논문집, pp.1-3, 2017

[8] Cicero dos Santos, Ming Tan, Bing Xiang, Bowen Zhou “Attentive Pooling Networks”, IBM Watson, T.J. Watson Research Center, NY. USA, pp.1-4, 2016

[9] Yoon Kim, “Convolutional Neural Networks for Sentence Classification”, Association for Computational Linguistics, pp.1-6, 2014