키워드 결합 Self-Attention Network를 이용한 문서 분류

박동주, 안창욱*
광주과학기술원 전기전자컴퓨터공학부
e-mail : {toriving, cwan}@gist.ac.kr

Classifying Documents with Self-Attention Network Built on Input-Keyword Combination

Dongju Park, Chang Wook Ahn*
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

요 약

최근 머신러닝 및 딥러닝 알고리즘은 전통적인 방법론을 사용하던 자연어 처리 분야에서 널리 활용되고 있다. 특히 주어진 문서가 어떤 범주에 속하는지 구분하는 문서 분류에서는 Convolutional Neural Network(CNN) 및 Recurrent Neural Network(RNN)이 사용되어 기존의 방법보다 성능을 향상시켰다. 하지만 CNN의 경우 필터 크기로 데이터를 처리하기 때문에 긴 문장이나 관계가 있는 단어 간의 거리가 멀어 지면 성능이 좋지 않을 수 있다. RNN의 경우는 CNN의 문제점을 보완 할 수 있지만, 이 역시 한계가 존재한다. 본 논문에서는 이러한 문제점을 해결하기 위해 CNN 및 RNN 계열 네트워크를 사용하지 않고, RNN 계열 네트워크의 성능 향상을 위해 고안 된 Attention mechanism을 하나의 간단한 네트워크로 활용하는 방법을 제안한다. 또한, 이 네트워크는 입력 데이터의 키워드를 추출하여 기존 입력 데이터와 결합시킨 후 새로운 입력 데이터를 만들어서 사용한다. 직접 수집한 네이버 뉴스 데이터와 문서 분류 및 텍스트 분류에서 널리 사용되고 있는 AG's news corpus 데이터로 실험한 결과 키워드를 결합하는 방법은 그렇지 않은 방법보다 성능이 향상되었고, 따라서 키워드가 성능에 영향을 준다고 판단된다. 또한, 제안한 키워드 결합 Self-Attention Network는 기본적인 CNN 및 RNN 계열 네트워크와 비교하여 보다 나은 성능을 얻었다.

1. 서 론

문서 분류는 자연어 처리 분야에서 개체명 인식, 문서 요약, 기계 번역 등과 함께 가장 기본적이면서도 중요한 역할을 하고 있으며, 우리의 실생활에서도 쉽게 접할 수 있다. 문서의 수는 현대사회의 빠른 발전으로 인해 인터넷 및 우리 일상에서 기하급수적으로 증가하고 있으며, 그에 따라 적절한 문서 분류의 필요성은 계속 증가하고 있다. 이러한 필요성이 증대됨에 따라 문서를 분류하는 방법이 다양하게 발전되고 있다.

최근 자연어 처리 분야에서는 머신러닝 및 딥러닝 알고리즘을 사용하여 자연어 처리의 세부 분야들을 접근하고 있다. 특히 문서 분류 분야에서는 딥러닝 알고리즘인 Recurrent Neural Network(RNN) 기반 Long-Short Term Memory(LSTM) [1]와 Convolutional Neural Network(CNN) [2]를 이용하여 주어진 문서내의 텍스트의 특징을 추출한 후, 문서를 분류하는 방법으로 좋은 성능을 보여 주고 있다.

CNN 계열의 네트워크는 필터의 크기로 데이터를 압축하기 때문에 긴 문장이나 문서의 경우 멀리 떨어진 어절이나 음절과 같은 입력 데이터 간의 상관관계를 찾지 못한다. 반면, RNN이나 LSTM 계열 네트워크는 이를 보완 할 수 있지만 일정 길이 이상의 문장 및 문서에서는 CNN 계열 네트워크와 같이 한계가 존재 한다. 또한 이러한 한계는 입력 데이터에서 중요한 데이터를 놓치는 원인이 되기도 한다.

본 논문에서는 앞서 언급한 한계를 극복하기 위해 Attention mechanism [3] 기반의 Self-Attention Network(SAN)를 바탕으로 Term Frequency-Inverse Document Frequency(TF-IDF)를 이용하여 추출한 키워드를 입력 데이터에 결합하여 사용하는 모델을 제안한다. 제안된 모델은 직접 수집한 네이버 뉴스 데이터와 AG's news corpus 데이터를 이용하여 기본적인 RNN 및 CNN 계열 네트워크와 성능을 비교 및 검증한다.

2. 관련 연구

문서 분류의 방법론으로는 크게 전통적인 방법과 최근에 들어 많이 사용하는 머신러닝 및 딥러닝을 이용하는 방법이 존재한다.

대표적인 전통적인 방법 중 하나는 TF-IDF를 이용하는 방법이다. TF-IDF는 여러 문서들이 존재 할 때 각 문서에서 어떤 단어가 얼마나 중요한 지 나타내는 통계적 수치이며, 문서의 핵심어를 추출하거나 문서를 분류하는데 사용이 된다. 본 연구에서는 키워드 추출을 위해 사용한다. TF-IDF는 Term Frequency 부분과 Inverse Document Freq

uency 부분을 이용하여 만든다. 일반적인 TF-IDF 식은 다음과 같다.

$$tf(t,d) = 0.5 + \frac{0.5 \times f(t,d)}{\max\{tf(w,d) : w \in d\}} \quad (식\ 1)$$

$$idf(t,D) = \log\frac{|D|}{1 + |\{d \in D : t \in d\}|} \quad (식\ 2)$$

$$TF-IDF(t,d,D) = tf(t,d) \times idf(t,D) \quad (식\ 3)$$

d : 문서
t : 단어
$tf(t,d)$: Term Frequency
$idf(t,D)$: Inverse Document Frequency
$|D|$: 전체 문서 D의 수
$|\{d \in D : t \in d\}|$: 단어 t가 포함된 문서의 수

최근 연구들은 전통적인 알고리즘을 벗어나 머신러닝 및 딥러닝 알고리즘을 많이 사용하며, 그 중 RNN과 CNN 계열 네트워크들이 문서 및 텍스트 분류에 사용된다.

RNN은 연속적인 입력의 데이터들을 처리하는 네트워크로, 이전 단계들의 입력과 현재 단계의 입력들 간의 상관관계를 찾는데 유용하여 텍스트 분류와 같은 자연어 처리에서 주로 사용되는 네트워크이다 [4].

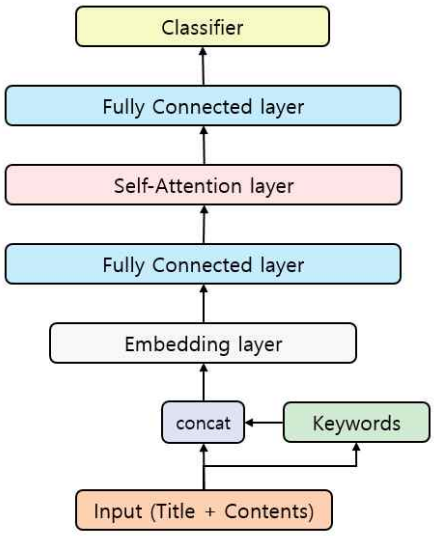
CNN의 경우 이미지 처리를 위한 네트워크로 처음 만들어졌지만, 최근에 들어 다양한 방법들을 활용하여 자연어 처리 분야에서도 널리 사용되고 있다. 그 중, 글자(character) 단위 [5]나 단어 단위 [6]로 데이터를 분리해 텍스트를 분류하는 연구들이 활발히 진행되고 있다.

기계 번역 분야에서는 Attention mechanism [3]을 통해 RNN 계열 네트워크의 정보 손실을 줄이고 기계 번역의 성능을 크게 향상시켰다. 또한, 최근 연구에서는 RNN 및 CNN을 사용하지 않고 Attention 구조로만 이루어진 네트워크 [7]가 등장하여 RNN과 CNN의 단점을 보완하였다. 이러한 Attention 기반 네트워크는 자연어 처리를 넘어 이미지처리 및 음성 신호처리분야에서도 중요한 보조 네트워크로 사용되어 Attention Mechanism의 중요성을 보여준다.

3. 키워드 결합 Self-Attention Network

제안한 모델의 전체적인 흐름은 입력데이터로부터 키워드를 추출 한 후 그 키워드와 입력데이터를 결합하여 네트워크의 입력으로 사용되며, 입력된 데이터는 Embedding layer, Self-Attention Layer를 거친 후 Classifier를 통해 분류를 한다. 전체적인 네트워크의 구조는 그림 1에서 볼 수 있다.

입력 데이터는 문서의 제목과 내용이며, 입력 데이터에 결합되는 키워드는 각 입력 데이터를 식 3을 적용하여 구한 값 중 높은 값을 가지는 5개의 단어들로 선정한다.



(그림 1) 전체적인 네트워크의 구조

선정된 키워드는 그림 2와 같이 [SEP] 토큰을 이용하여 입력 데이터와 결합한 후 최종 입력 데이터로 사용된다.

(그림 2) 키워드를 결합한 입력 데이터

이렇게 구해진 최종 입력 데이터는 Neural Network에서 사용되기 위해 Embedding layer를 통해 벡터로 변환하고, Fully Connected layer에 입력 값이 된다. Self-Attention layer에서는 Fully Connected layer의 출력 값을 입력으로 받아 선형 함수를 한번 거친 후 식 4와 같은 연산을 한다.

$$\text{Attention}(V) = \text{softmax}(\frac{VV^T}{\sqrt{d_v}})V \quad (식\ 4)$$

식 4에서 V는 선형 함수에서 나온 출력 값을 나타내며 d_v는 V의 차원을 뜻한다. 그 후 연산 된 값을 최종적으로 선형 함수를 한 번 더 통과시킨다. 이러한 Self-Attention layer는 하나의 입력 데이터 안에서 어떤 데이터가 출력으로 나오는 결과에 어느 정도 영향을 미치는지 계산하는 역할을 한다. Self-Attention layer 연산 이후 Fully Connected layer를 통과 한 후 Classifier를 통해 문서의 범주가 무엇인지 분류를 한다.

Attention 연산 전 선형함수를 제외 한 모든 선형함수와 Fully Connected layer 후에는 Dropout [8]과 Layer normalization [9]을 적용시킨다. 또한 Self-Attention layer에서는 Layer normalization 전에 Residual Connection [10]을 추가 하였다. 전체적으로 Attention mechanism을 네트워크에 포함하고 기존에 있던 Attention 계열 네트워크를 간소화 했다.

4. 실험 및 결과

4.1. 실험 설정

학습 및 검증 데이터 본 논문에서는 학습 및 검증 데이터로 네이버 뉴스 데이터와 AG's news corpus 데이터 두 가지를 사용하였다. 네이버 뉴스 데이터는 매일 30개씩 갱신되는 네이버 랭킹뉴스 페이지[1]에서 정치, 경제, 사회, 생활/문화, 세계, IT/과학 총 6가지 범주를 선택하였으며, 비슷한 뉴스가 수집되는 것을 막기 위해 2019년 3월 10일 기준으로 5일 간격으로 기사의 제목과 네이버 뉴스에서 제공하는 기사 내용의 요약을 수집하였다. 각 범주 당 5000개의 데이터를 수집하여 총 30000개의 데이터로 이루어져 있으며, 학습데이터로 70%, 검증데이터로 15%, 테스트데이터로 15%를 사용하였다.

AG's news corpus는 [5]에서 사용된 데이터로 4가지 갈래로 이루어져 있으며, 각 범주는 30000개의 학습 데이터와 1900개의 테스트 데이터로 이루어져 있다. 학습데이터 중 90%는 학습데이터로 사용하고 10%는 검증데이터로 사용하였다.

데이터 전처리 네이버 뉴스 데이터의 경우 제목과 내용에서 명사만 추출하여 사용하였으며, AG's news corpus는 띄어쓰기 단위로 분절하여 하나의 단어로 취급하였다.

두 데이터 공통으로 각 데이터들의 평균길이를 최대 길이로 설정하였으며, 길이가 초과되는 데이터들은 초과되는 부분을 제거하고, 부족한 데이터들은 [PAD] 토큰으로 채워 주었다. 제목과 키워드 그리고 내용 순으로 [SEP] 토큰을 사이에 추가한 뒤 결합하여 하나의 데이터로 만들었다. 또한 단어 사전의 경우 빈도수가 높은 순으로 상위 30000개의 단어를 사용하였으며, 그 외 단어들은 [UNK] 토큰으로 대체하였다.

하이퍼파라미터 네이버 뉴스 데이터에서는 learning rate를 0.0005로 사용하였고, AG's news corpus 데이터에서는 0.0001을 사용하였다. 두 데이터 모두 Adam Optimizer를 사용하였으며, 배치 사이즈는 128, 입력 Embedding 사이즈는 256, 모든 레이어의 hidden size는 512, Dropout rate는 0.5로 고정하였다. 비교 모델 중 CNN의 경우 총 3개의 필터를 사용하였으며, 필터 사이즈는 각각 3, 4, 5로 정하였다. 또한 CNN내의 모든 필터의 개수는 128개로 정하였다.

비교 모델 본 연구에서는 총 3가지 모델과 비교하였다. 첫 번째는 RNN계열 모델인 Bi-directional LSTM모델을, 두 번째는 CNN모델, 그리고 마지막으로 키워드를 결합하지 않은 Self-Attention 모델이다. 네이버 뉴스 데이터의 경우 모든 모델을 직접 구현 및 평가를 하였고, AG's news corpus의 경우는 [5]에서 결과를 참조하였다.

4.2. 실험 결과

실험 결과 네이버 뉴스 데이터와 AG's news corpus 데이터 모두 키워드를 결합한 SAN이 가장 높은 성능을 보였다. 네이버 뉴스 데이터의 경우 키워드를 결합하지 않은 SAN보다 CNN이 더 높게 나왔지만 AG's news corpus의 경우에는 키워드를 포함하지 않은 SAN이 더 높게 나왔다. 이를 바탕으로 데이터에 따라 키워드를 결합하지 않은 SAN과 CNN과의 성능우위가 달라질 수 있음을 시사한다. 또한 키워드를 결합하지 않은 것 보다 키워드를 결합한 SAN이 성능이 더 높다는 점에서 결합된 키워드가 성능에 영향을 주고 있음을 알 수 있다. 실험 결과는 데이터별로 표 1과 표 2에 나타나 있다.

(표 1) 네이버 뉴스 데이터에 대한 모델별 분류 정확도

모델	분류 정확도
Bi-LSTM	82.11%
CNN	83.97%
SAN w/o keywords	83.73%
SAN with keywords	84.22%

(표 2) AG's news corpus에 대한 모델별 분류 정확도

모델	분류 정확도
Bi-LSTM	86.06%
CNN	90.49%
SAN w/o keywords	90.95%
SAN with keywords	91.32%

5. 결 론

본 논문에서는 최근 자연어 처리 분야에서 주목을 받고 있는 Attention mechanism을 활용하여 기존에 있던 Attention 계열 네트워크를 간소화한 키워드 결합 Self-Attention Network를 제안하였다. 제안 모델은 문서에서 키워드를 추출해 입력데이터와 결합한 후, SAN을 이용하여 문서 분류를 한다. 실험 결과 기본적인 RNN 및 CNN 계열 네트워크에 대해서 네이버 뉴스 데이터와 AG's news corpus 데이터에서 좋은 성능을 나타냈다. 또한 문서 및 텍스트 분류에서 키워드를 이용하여 성능을 향상 시킬 수 있다는 점을 보여주었다.

또한, 제안된 모델은 문서분류 뿐만 아니라 질의응답(QA) 및 문서 요약과 같은 자연어 처리 분야와 더 나아가 다양한 딥러닝을 활용하는 분야에서 성능의 향상을 기대 할 수 있을 것이다.

Acknowledgment

이 논문은 2019년도 광주과학기술원의 재원으로 GRI(GIST연구원) 사업의 지원을 받아 수행된 연구임

1) https://news.naver.com/main/ranking/popularDay.nhn?mid=etc&sid1=111

참 고 문 헌

[1] Hochreiter, S., & Schmidhuber, J. "Long Short-Term Memory", Neural Computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735. 1997.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. "Imagenet classification with deep convolutional neural networks", In NIPS. 2012.

[3] Bahdanau, D., Cho, K., & Bengio, Y. "Neural machine translation by jointly learning to align and translate", arXiv preprint arXiv:1409.0473. 2014.

[4] Liu, P., Qiu, X., & Huang, X. "Recurrent neural network for text classification with multi-task learning", arXiv preprint arXiv:1605.05101. 2016.

[5] Zhang, X., Zhao, J., & LeCun, Y. "Character-level convolutional networks for text classification", In NIPS.. 2015.

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

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Polosukhin, I. "Attention is all you need", In NIPS. 2017.

[8] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. "Dropout: a simple way to prevent neural networks from overfitting", The Journal of Machine Learning Research, 15(1), 1929-1958. 2014.

[9] Ba, J. L., Kiros, J. R., & Hinton, G. E. "Layer normalization", arXiv preprint arXiv:1607.06450. 2016.

[10] He, K., Zhang, X., Ren, S., & Sun, J. "Deep residual learning for image recognition", Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.