

# CNN과 LSTM 및 GRU 기반 연구 논문 분류 시스템의 설계 및 구현

비스와스 디프토\*, 강지훈\*\*, 길준민\*\*\*1)

\*대구가톨릭대학교 컴퓨터소프트웨어학과

\*\*고려대학교 4단계 BK21 컴퓨터학교육연구단

\*\*\*대구가톨릭대학교 컴퓨터소프트웨어학부

\*dipto.biswas94@gmail.com, \*\*k2j23h@korea.ac.kr, \*\*\*jmgil@cu.ac.kr

## Research Paper Classification Scheme based on CNN with LSTM and GRU

Biswas Dipto\*, Jihun Kang\*\*, Joon-Min Gil\*\*\*

\*Dept. of Computer Software Engineering, Daegu Catholic University

\*\*BK21 FOUR R&E Center for Computer Science and Engineering, Korea University

\*\*\*School of Computer Software Engineering, Daegu Catholic University

### 요 약

최근 딥러닝 기술은 자연어처리에서 기본적이고 필수적인 기법으로 자연어처리에 필요한 복잡한 비선형 관계를 모델링할 수 있다. 본 논문에서는 LSTM(Long Short-Term Memory)과 GRU(Gated Recurrent Unit) 딥러닝 기술을 연구 논문 분류에 적용하며, CNN(Convolutional Neural Network)에 LSTM과 GRU를 각각 결합하여 특정 분야의 연구 논문을 분류하고 연구 논문을 추천하는 기법을 제안한다. 워드 임베딩과 딥러닝 기법을 연구 논문 분류에 적용하여 관심이 있는 단어와 단어 주변의 단어 들 사이의 유사성과 성능을 비교 분석한다.

### 1. 서론

텍스트 분석은 네트워크 환경에 대한 여론 모니터링, 서비스 평가 및 만족도 분석 작업 등 다양한 분야에 적용되어 왔다[1]. 현재까지 효과가 좋은 텍스트 분석 알고리즘은 대부분 통계적 학습 방법에 기반하고 있다. 최근 딥러닝 기술은 텍스트 분석 등 자연어 처리에서 기본적이고 필수적인 기법으로 자연어처리에 필요한 복잡한 비선형 관계를 모델링할 수 있다.

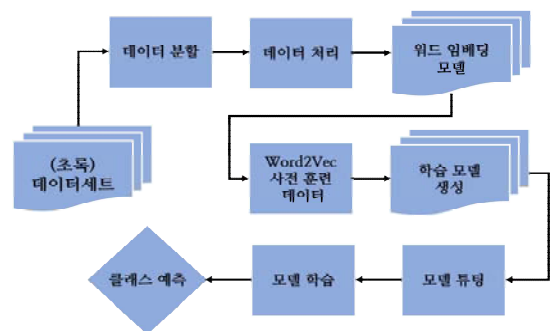
현재 딥러닝 기술 중 CNN과 RNN은 텍스트 분석 작업에 좋은 성능을 보여왔다. 실제 텍스트 분석 작업에서는 RNN보다 CNN이 좋은 성능을 보이고 있지만, 우수한 텍스트 분석을 위해 CNN과 RNN 모델을 결합한 모델도 제안되었다[2]. 이들 모델을 통해 스팸 메일 분류, 쇼핑 및 영화 리뷰 감성 분류, 뉴스 분류 등 다양한 텍스트 분석 작업이 진행되어 왔다.

본 논문에서는 RNN의 기본적인 방법인 장단기 메모리(Long Short-Term Memory: LSTM)와 게이트 순환 유닛(Gated Recurrent Unit: GRU)을 각각 연구 논문의 분류에 적용하고, CNN에 LSTM과 GRU를 각각 결합한 모델을 연구 논문 분류에 적용하고 각각의 성능을 비교 및 분석하고자 한다.

### 2. 시스템 흐름도

본 논문의 연구 논문 분류 기법에 관한 전반적인

흐름은 (그림 1)과 같다.



(그림 1) 연구 논문 분류 기법의 전반적인 흐름

(그림 1)에서 논문 정보에 해당하는 데이터셋은 “Science Direct” 웹사이트에서 제공하고 있는 FGCS(Future Generation Computer Systems) 저널의 초록 부분을 웹크롤링하여 수집하고 활용한다. 다음으로, 데이터 분리 및 전처리 작업을 수행한다. 그런 다음, 워드 임베딩과 데이터 사전 훈련 기반으로 딥러닝 모델을 학습한다. 마지막으로 논문의 초록 데이터를 사전 훈련된 학습과 테스트 데이터를 기반으로 성능평가를 수행하고 결과를 도출한다.

### 3. 데이터 처리

약 5,659개의 연구 논문에서 초록(abstract) 데이터를 수집하여 논문 데이터셋을 구성하였다. 다음으

1) 교신저자

로, 수집된 데이터를 Gensim 라이브러리[3]를 활용하여 토큰화를 수행하며 이를 통해 단어 단위로 분할을 한다. 그런 다음, 문장 내의 단어들 중에 URL, 숫자, 구두점, 불용어 등 불필요한 단어들을 제거한다. 또한, 워드 임베딩 작업 준비를 위해 각 단어들을 명사 형으로 변환한 후, 워드 임베딩 작업을 수행한다.

#### 4. 워드 임베딩

본 논문에서는 word2vec 기반의 워드 임베딩 방식[4]을 사용한다. word2vec은 2-계층 신경망을 사용하여 단어 임베딩을 학습하는 가장 널리 사용되는 기술 중 하나이다. 한편, 워드 임베딩은 텍스트 분류를 위해 word2vec 형태로 개발되었으며, 대표적 방식으로 Mikolov[5]가 제안한 2가지 학습 방법인 CBOW(Continuous Bag of Words)와 Sg(Skip-gram)가 있다. 이들 방법의 주요 차이점은 CBOW가 컨텍스트를 사용하여 대상 단어를 예측하는 반면, Sg는 단어를 사용하여 대상 컨텍스트를 예측한다.

본 논문에서는 연구 논문의 초록에 있는 모든 단어를 사용하여 word2vec 모델을 구성하였다. 이를 위해 임베딩의 차원 수는 100으로 설정하였다. 대상 단어와 대상 단어 주변의 단어 사이의 최대 거리를 계산하기 위해 윈도우 크기를 5로 설정하였으며, 1회 이상 나오는 단어만을 대상으로 100번의 반복을 수행하였다. 한편, 워커(worker)의 수는 3으로 설정하였다. 이런 설정에 기반하여 훈련 알고리즘으로 CBOW와 Sg를 word2vec 모델에 각각 적용하였다.

#### 5. 제안 모델 구조

##### 5.1 LSTM 모델 구조

LSTM은 RNN에 비해 더 효과적인 방식으로 장기 종속성을 유지하는 특수한 유형의 RNN이다[6]. LSTM은 (1) 입력 게이트, (2) 삭제 게이트, (3) 셀 상태, (4) 출력 게이트와 은닉 상태를 사용하여 각 노드 상태에 허용될 정보의 양을 신중하게 조절하기 때문에 기울기 소실 문제를 극복하는데 특히 유용하다.

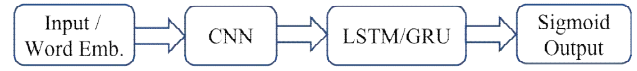
##### 5.2 GRU 모델 구조

GRU[7]는 LSTM 아키텍처를 단순화한 모델이다. GRU에서는 업데이트 게이트와 리셋 게이트 두 가지 게이트만이 존재한다. GRU는 LSTM과 비슷하지만 출력 게이트가 없으므로 LSTM보다 매개변수가 적다. 또한, LSTM보다 학습 속도가 빠르다고 알려졌지만 대부분의 평가나 작업에서 비슷한 결과를 보여 왔다.

##### 5.3 CNN과 LSTM, GRU 모델 구조

최근 들어, RCNN(Recurrent Convolutional Neural Network) 또한 텍스트 분류에 적용되고 있다. 이 기술의 주요 아이디어는 반복 구조로 컨텍스트 정보를 캡처하고 컨볼루션 신경망을 사용하여 텍스트 표현을 구성하는 것이다. 이 아키텍처는 두 가

지 기술의 장점을 모두 사용하기 위해 RNN과 CNN을 결합한 것이다. (그림 2)는 CNN에 LSTM 혹은 GRU를 결합한 모델의 구조를 보여준다.



(그림 2) CNN에 LSTM 혹은 GRU의 결합 모델의 구조

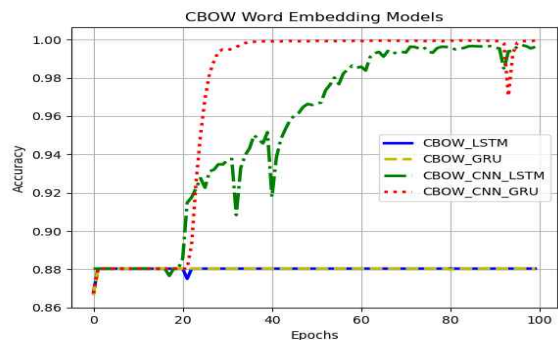
본 논문에서는 CNN에 LSTM 혹은 GRU를 결합한 모델을 각각 활용하여 연구 논문의 분류를 위해 다음과 같은 작업을 수행한다. RCNN 임베딩 레이어 뒤에 다차원 CNN과 맥스 풀링 레이어를 추가하였다. 길이가 2, 3, 4인 필터가 사용되었으며, 각각은 128개의 세트에 구성하였다. LSTM과 GRU 각각 레이어에 100개 메모리를 적용하였고, 그 뒤에 리턴 시퀀스와 순환 드롭아웃을 설정하였다. 그런 다음, 모델 체크포인트와 콜백을 사용하여 검증 정확도가 최대일 때 모델 가중치를 저장하도록 하였다. 풀링 레이어는 표준 길이 2를 사용하여 특징 맵의 크기를 절반으로 줄였다.

본 논문에서는 FGCS 저널 논문의 초록 데이터에 대해서 텐서플로우[8]에서 제공되는 토큰화 라이브러리를 사용하여 단어별로 토큰화를 수행하였으며, 각 논문마다 단어 개수가 서로 다르기 때문에 일률적으로 154개로 설정하여 임베딩 레이어에 적용하였다. 그리고 워드 임베딩 기법에 적용할 단어는 초록에서 50번 이상 출현한 단어 중에 가장 많이 나오는 10개 단어만을 추출하였다. 이들 10개 단어에 대해서 특정한 단어가 초록에서 5번 이상 나오면 그 단어가 해당 논문과 연관성이 있다고 판단하였으며, 이를 근거로 학습 데이터를 생성하였다.

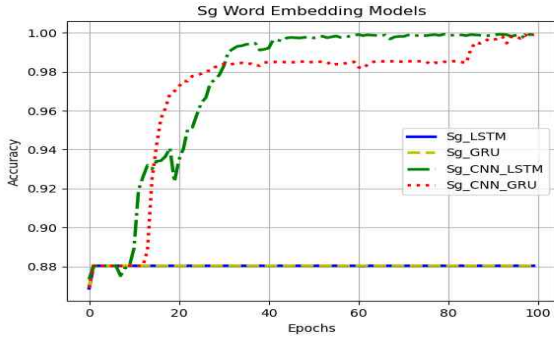
##### 5.4 모델 학습 및 테스트 결과

본 논문에서는 CBOW와 Sg 임베딩을 기반으로 LSTM과 GRU 모델을 사용하여 텍스트 분류를 수행하며, CNN에 LSTM과 GRU를 각각 결합한 모델에 대해서도 텍스트 분류를 수행하였다.

학습 반복수(epoch)에 대한 CBOW와 Sg 모델 각각의 학습 정확도는 (그림 3)과 (그림 4)에서 보여준다.



(그림 3) CBOW 기반 모델의 정확도



(그림 4) Sg 기반 모델의 정확도

(그림 3)과 (그림 4)의 결과를 분석하면, CBOW와 Sg 기반 모델 모두에 대해서 LSTM과 GRU를 사용하는 CNN 모델이 다른 모델에 비해서 높은 정확도를 보여주었다. 한편, (그림 3)의 결과로부터 CBOW의 경우 GRU를 사용하는 CNN 모델이 좀 더 빠르게 수렴함을 알 수 있으며, (그림 4)의 결과는 Sg의 경우 LSTM을 사용하는 CNN 모델이 다른 모델에 비해 좀 더 빠르게 수렴함으로 보여주었다.

### 6. 실험 결과

제안 기법의 성능을 측정하기 위해 정밀도 (precision)와 재현율(recall)에 기반한 F-Score[9]를 사용하였다.

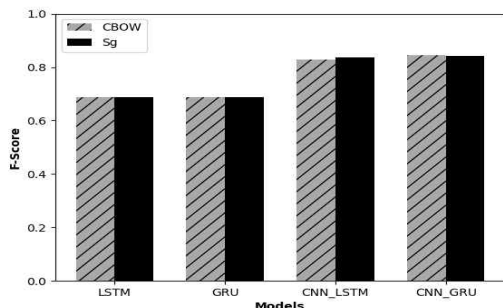
$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$F\text{-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (3)$$

여기서, TP, FN, FP는 각각 True Positive, False Negative, False Positive를 의미한다.

한편, scikit-learn 라이브러리[10]를 활용하여 수집된 연구 논문 데이터를 학습 데이터셋과 테스트셋으로 분리하였으며, 이들 데이터셋을 활용하여 모델의 성능평가를 수행하였다.



(그림 5) 4가지 모델에 대한 F-Score 결과

(그림 5)는 CBOW와 Sg 임베딩을 기반으로 LSTM,

GRU, 그리고 LSTM과 GRU를 사용하는 CNN 모델의 F-Score 결과를 보여준다. (그림 5)의 결과를 분석하면, 4가지 모델 중에서 CNN에 LSTM과 GRU이 각각 결합된 모델에 대해서 Sg 기법이 CBOW 기법에 비해서 높은 F-Score 값을 보여주었다. 한편, 워드 임베딩에 기반한 Sg 기법은 CBOW 기법보다 좀 더 좋은 성능을 가짐을 알 수 있다.

### 7. 결론

본 논문에서는 워드 임베딩 기법을 적용하여 LSTM과 GRU를 사용하는 CNN 기반의 연구 논문 분류를 바탕으로 연구 논문의 추천 방법을 제안하였다. 4가지 모델인 CNN, LSTM, 그리고 CNN에 LSTM과 GRU를 각각 결합한 모델을 연구 논문의 분류에 적용하였으며, F-Score를 활용하여 이들 모델의 성능을 비교 및 분석하였다. CNN에 LSTM과 GRU를 결합한 모델은 다른 모델에 비해서 비교적 높은 정확도를 보여주었으며, 성능평가 결과는 워드 임베딩 기반한 Sg 기법이 CBOW 기법에 비해 높은 성능을 가짐을 보여주었다.

### Acknowledgment

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2022R1A2C1092934).

### 참고문헌

- [1] M. Hu, and B. Liu, "Training and Summarizing Customer Reviews," Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 168-177, Aug. 2004.
- [2] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Random Multimodel Deep Learning for Classification," Proc. of 2nd Int. Conf. on Information System and Data Mining ICISDM' 18., pp. 9-11, April 2018.
- [3] Gensim, <https://pypi.org/project/gensim/>
- [4] word2vec, <https://radimrehurek.com/gensim/index.html>
- [5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Proc. of the Int. Conf. on Learning Representations (ICLR), 2013.
- [6] S. Hochreiter and, J. Schmidhuber, "Neural Computation," Military Applications of Artificial Intelligence: Ethical Concerns in an Uncertain World, vol. 9, iss. 8, pp. 1735-1780, Nov. 1997.
- [7] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," Proc. of SSST-8, Proc. of 8th Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103-111, 2014.
- [8] Module: tf.optimizers | TensorFlow Core v2.7.0, TensorFlow, Retrieved Nov. 6, 2021.
- [9] David M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," pp. 37-63, 2011.
- [10] scikit-learn: A set of python modules for machine learning and data mining, retrieved Sept. 24, 2022.