Journal of The Korea Society of Computer and Information
Vol. 23 No. 11, pp. 31-41, November 2018

www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.11.031

A Deeping Learning-based Article- and Paragraph-level Classification

Euhee Kim*

Abstract

Text classification has been studied for a long time in the Natural Language Processing field. In this paper, we propose an article- and paragraph-level genre classification system using Word2Vec-based LSTM, GRU, and CNN models for large-scale English corpora. Both article- and paragraph-level classification performed best in accuracy with LSTM, which was followed by GRU and CNN in accuracy performance. Thus, it is to be confirmed that in evaluating the classification performance of LSTM, GRU, and CNN, the word sequential information for articles is better than the word feature extraction for paragraphs when the pre-trained Word2Vec-based word embeddings are used in both deep learning-based article- and paragraph-level classification tasks.

▸Keyword: Genre Classification, Deep Learning, Word2Vec, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN), Word embedding

I. Introduction

장르 분류 문제는 자연어 처리 분야에서 오랫동안 연구되어 왔다. 그 응용분야는 사용 언어와 처리할 텍스트 데이터의 크기가 커짐에 따라 더욱 중요성이 강조되고 있다.

현대 미국 영어 말뭉치 (The Corpus of Contemporary American English: COCA)나 세종 말뭉치와 같이 각 언어를 대표하는 대용량의 말뭉치는 언어 사용의 대표성을 균형적으로 반영하기 위해 다양한 장르에서 생산된 텍스트로 구성되어 있다. 텍스트마다 태그(tag)되어 있는 장르 정보를 이용하여 텍스트를 분류하고 있어 언어 사용 특성을 비교하고자 하는 코퍼스 연구에서 유용하게 활용되고 있다. 장르 기반을 둔 텍스트 분류 방식은 코퍼스 구축 시 고려되었던 장르 정보에만 의존한 것으로 텍스트 분류하고 비교하는데 한계가 있다[1, 2, 3].

데이터마이닝 분야에서 코퍼스를 대상으로 군집분석 (cluster analysis)의 통계 모델링을 이용하여 자연어 문법 형식의 분포와 관련한 텍스트 유형의 관점에서 텍스트를 분류하고 있다[4].

최근 기계학습 분야에서 말뭉치의 장르 기반을 둔 텍스트의 어휘 분포 정보를 이용하여 텍스트 사이에 존재하는 일련의 주제(topic)를 통계적 방법으로 어휘별 그리고 텍스트별 분포 정보를 요약하는 토픽모델링이 텍스트 분류 문제에 많이 사용되고 있다[5].

또한, 단어의 빈도수를 특징으로 사용한 Bag-of-Words (BOW) 방법과 단어의 의미적, 문맥적 정보를 파악하기 위해 N-gram 방법을 사용하여 문장을 분류하고 있다[6, 7]. 그러나 N값이 커질수록 계산량이 많아지는 단점이 있다.

이러한 단점을 해결하기 위해 딥러닝 분야에서 많이 사용하고 있는 컨볼루션 신경망을 이용하여 커널을 통해 특징을 추출하고 문맥 정보를 파악하기 위해 커널의 크기를 변화시켜 대용량의 문장을 분류하였다. 또한 언어모델링에서 순차적인 데이터 처리에 뛰어난 성능을 내고 있는 순환 신경망을 이용해서 인터넷에서 수집한 뉴스 장르인 텍스트를 대분류 및 소분류로 분류하였다[8, 9, 10].

본 논문에서는 대용량의 영어 말뭉치 COCA을 딥러닝 모델로 학습하여 기사 (article) 또는 문단 (paragraph)에 대한 장르 (genre)를 분류하는 시스템을 제안하는 것을 목적으로 한다. 이를 위해 BOW 또는 N-gram을 이용하는 대신 텍스트에서 문맥 정보를 파악하기 위해 워드 임베딩 (word embedding) 기술을 적용하여 순차적인 언어 데이터 처리와 특징 정보를 추출하여 장르를 분류하기 위해 장기-단기 기억 신경망, 회로형 순환 유닛, 그리고 컨볼루션 신경망을 사용한다.

본 논문의 2장에서는 장르 분류에 사용되는 딥러닝 모델을

• First Author: Euhee Kim, Corresponding Author: Euhee Kim
*Euhee Kim (euhkim@shinhan.ac.kr), Computer Science & Engineering, Shinhan University
• Received: 2018. 10. 22, Revised: 2018. 11. 09, Accepted: 2018. 11. 10.
• This work was supported by the Shinhan University Research Fund 2018.

소개하며, 3장에서는 말뭉치의 기사 또는 문단을 입력 단위로 장르를 분류하는 딥러닝 분류 시스템을 제안하며, 4장에서는 제안한 분류모델의 성능을 비교하고 분석한다. 5장에서는 실험 결과에 대한 결론을 맺는다.

II. Related Works

1. Word2Vec

Word2Vec은 벡터공간에서 비정형 데이터인 단어의 표현 방법을 구현한 알고리즘이다. 각 단어를 공간에서 연속벡터로 표현하고 두 단어 간의 거리는 관계성을 나타내고 두 단어를 이은 방향은 문맥상의 의미를 내포하게 된다.

자연언어 처리 기계학습 분야에서 Word2Vec은 같은 맥락을 지닌 단어는 가까운 의미를 지니고 있다는 전제에서 출발한다. 방대한 양의 텍스트 문서를 통해 Word2Vec 모델을 사전 학습 진행하며 인공신경망에 워드 임베딩 하여 학습시킨다. 연관된 의미의 단어들은 문서상에 가까운 곳에 출현할 가능성이 높기 때문에 학습을 반복해 나가는 과정에서 두 단어는 점차 가까운 벡터를 지니게 된다.

Word2Vec은 BOW와 N-gram 방법을 통해 단어의 문맥상의 정보를 계산하는데 급격이 늘어나는 계산량을 줄이고 분석 결과의 정확도를 비약적으로 향상시켰다[11, 12].

2. Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNN)은 히든 노드가 방향을 가진 엣지으로 연결되고 순환구조를 이루는 인공신경망 모델이다. 이전 계산 결과에 대한 메모리 정보를 이용하여 순차적으로 등장하는 텍스트 처리에 적합한 모델이다 [13].

RNN은 문장 길이에 관계없이 입력과 출력을 받을 수 있는 네트워크 구조이며 역전파 (backpropergation)을 이용하여 오차를 최소화하는 방향으로 파라미터 값들을 갱신하면서 학습해 간다. 그러나 관련 단어와 그 단어를 사용하는 지점 사이의 거리가 멀 경우 역전파 수행 시 학습 능력이 크게 저하되는 vanishing gradient 문제를 갖고 있다. 이런 문제를 극복하기 위해 RNN의 변형으로서 고안된 모델이 LSTM이다.

LSTM은 긴 순차적인 정보를 회로 forget gate와 input gate를 통해 과거 정보를 잊거나 현재 정보를 기억할 수 있도록 하여 vanishing gradient 문제를 완화시켜 성능을 크게 향상시킨다[14].

3. Gated Recurrent Unit (GRU)

GRU은 LSTM의 장점을 유지하면서 파라미터 수를 줄여서 LSTM보다 학습속도가 더 빠르고 계산복잡성을 낮춘 RNN의 변형 학습모델이다. GRU도 vanishing gradient 문제를 극복했다는 점에서 LSTM과 유사하지만 게이트 일부를 생략한 형태이다. update gate와 reset gate를 통해 현 시점 정보와 과거 직전 시점 정보를 사용하면서 과거 정보를 얼마나 반영할지를 결정하여 갱신한다[15, 16].

4. Convolutional Neural Networks (CNN)

CNN은 컴퓨터 비젼 분야에서 이미지 분류에 뛰어난 성능을 보여주는 인공신경망 모델이다. Convolution 층과 max-pooling 층을 번갈아 가며 학습을 수행하고 마지막에는 fully-connected 층을 이용하여 분류를 수행한다[17].

CNN 모델은 특징 추출 문제에 뛰어난 성능을 보여 왔으며 최근에는 자연어 처리의 응용 분야에서도 널리 활용되고 있다. 텍스트 분류 작업에 대한 CNN모델은 이미지 픽셀 대신 문장이나 문서들을 벡터로 변환한 후 해당 벡터 값들을 나열해서 2차원 이미지 행렬처럼 만든다. 문자 기반이 아니라 단어 기반으로 특징 데이터를 정한다. 행렬의 각 행은 하나의 단어를 벡터로 표현하며 One-hot-encoding 또는 Word2Vec을 이용한 저차원 (low-dimension) 워드 임베딩 벡터가 된다. 이 행렬은 결국엔 이미지 데이터와 같은 입력 양식을 가지게 되고 해당 값을 입력으로 CNN을 돌려서 분류 학습을 수행한다[18, 19, 20, 21].

III. Deep Learning-based Article- and Paragraph-level Genre Classification System

3장에서는 딥러닝 기술을 활용하여 기사 단위 또는 문단 단위로 장르를 분류하는 시스템을 제안한다. 텍스트 자동 분류에 필요한 텍스트 전 처리, 딥러닝 학습 방법, 그리고 학습한 딥러닝 모델의 적용 방법을 소개한다.

1. The Outline of the Deep Learning-based Genre Classification System

기사는 하나 이상의 문단들로 구성되고, 문단은 하나 이상의 문장들로 구성된다. 문장은 하나 이상의 단어로 구성된다. 문장에서 명사 단어는 분류에 필요한 핵심 정보를 포함한다. 따라서 기사와 문단을 명사들로 정의하고 Academic, Fiction, Magazine, Newspaper 등의 장르로 자동 예측하는 딥러닝 기반 분류 시스템은 그림 1과 같이 장르 분류 학습 단계와 장르 예측 단계로 나뉜다.

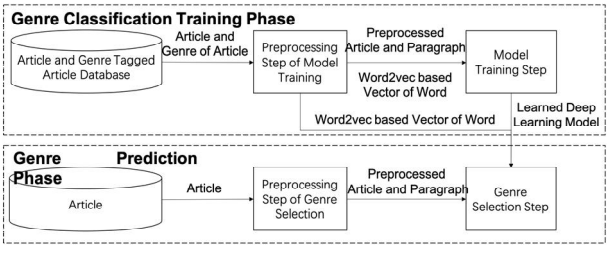

Fig. 1. Deep Learning-based Classification System

장르 분류 시스템의 학습 단계는 다음과 같다. 첫 번째, 전처리 단계에서는 장르를 분류하는데 사용되는 명사 단어를 추출하고 해당 단어를 Word2Vec 모델을 이용하여 벡터로 생성한다. 두 번째, LSTM, GRU, CNN과 같은 딥러닝 모델을 사용하여 장르 분류학습을 진행한다.

장르 분류 시스템의 예측 단계는 다음과 같다. 첫 번째, 전처리 단계에서는 학습된 딥러닝 모델로 장르를 예측하기 위한 테스트 명사 단어들을 추출한다. 두 번째, 전 처리된 기사와 문단 그리고 학습된 모델을 이용하여 장르를 예측한다.

2. The Preprocessing of Model Training

일반적으로 자연어 처리 과정에서 텍스트를 분류할 때 불필요한 대명사, 접속사, 관사 등과 같은 불용어 및 특수문자 제거 등과 같은 전 처리가 필요하다.

학습 모델의 전 처리 과정에서 다음과 같은 사항을 고려하였다. 첫 번째, 형태소 분석을 통해 명사 및 고유명사만을 추출하기 때문에 별도로 불용어 제거가 필요하지 않았다. 두 번째, 하나 이상의 문장으로 구성된 문단들을 고려하였다. 세 번째, 문장의 시작 및 종료 부분에 <BOS> 및 <EOS>로 표시할 경우에 언어모델과 달리 장르 분류 시 성능이 감소하기 때문에 구분하지 않았다. 네 번째, “.”, “ : ”, “ / ”, “@” 등과 같은 특수기호 제거가 필요하나 명사 및 고유명사 추출로 인해 제거되기 때문에 특수기호 제거가 별도로 필요하지 않았다.

모델 학습의 전 처리 과정은 그림 2와 같이 처리된다. 말뭉치의 기사에 해당 장르를 지정한다.

Article List and Article Tagging List Generation에서는 각 장르마다 중복해서 출현하는 특수 단어들은 분류 성능에 영향을 미치기 때문에 사용 횟수를 1회로 처리한다. 예를 들어, Academic 장르로 분류된 기사는 “section”과 같은 단어가 중복해서 사용되는 구조이여서 사용 횟수를 1회로 제안하였다. 말뭉치의 기사들을 식별하기 위해 기사 리스트를 생성한다. 각 기사에 장르를 표시하여 기사 태깅 리스트를 생성한다.

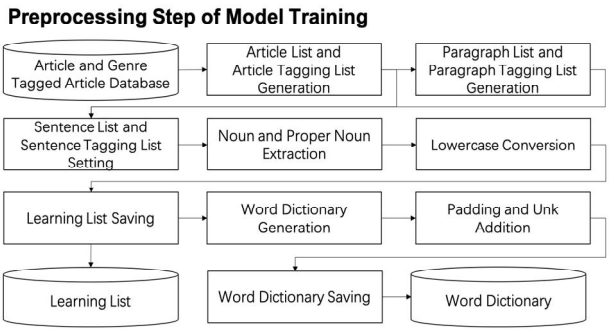

Fig. 2. The Preprocessing of Model Training

Paragraph List and Paragraph Tagging List Generation에서는 각 기사를 문단 단위로 분할하여 문단 리스트를 생성한다. 각 문단의 장르는 해당 기사의 장르로 표시하고, 문단 태깅 리스트를 생성한다. 문단으로 기사의 장르를 분류하는 경우에 수행하였다. 기사 단위로 분류하는 경우 학습에 사용되는 문장의 개수가 부족한 점을 해결하기 위해 기사를 문단으로 분할하여 활용하였다.

Sentence List and Sentence Tagging List Setting에서는 기사의 장르 분류에 사용하는 문장 리스트와 태깅 리스트를 설정한다. 기사를 사용하는 경우에는 문장 리스트 및 태깅 리스트를 기사 리스트 및 기사 태깅 리스트로 설정하였다. 문단을 사용하는 경우에는 문장 리스트 및 태깅 리스트를 문단 리스트 및 문단 태깅 리스트로 설정하였다.

Noun and Proper Noun Extraction에서는 형태소 분석을 통해 문장에서 명사 또는 고유명사 품사를 갖는 단어들을 추출하여 문장 리스트를 구성한다. 예를 들어, 영어 단어의 명사 품사 종류로는 보통명사 (NN), 복수 명사 (NNS), 고유명사 (NNP), 그리고 복수 고유명사 (NNPS) 기반으로 명사 및 고유명사를 추출하였다.

Lowercase Conversion에서는 문장 리스트에 있는 명사 및 고유명사 단어들을 소문자로 변경한다. 영어 문장의 첫 번째 단어 및 대문자로 작성된 고유명사들은 문장 내의 다른 단어들과 대소문자 구분이 불필요하므로 기사 리스트와 문단 리스트에 포함된 모든 단어들을 소문자로 변경하였다.

Learning List Saving에서는 문장 리스트에 태킹 리스트를 저장한다. 태깅 리스트를 활용하여 문장 리스트의 문장들을 장르별로 통합한다. 통합 후에 학습 리스트를 저장한다.

Word Dictionary Generation에는 Word2Vec 모델 학습을 통해 단어 사전을 생성한다. 문장 리스트를 이용하여 Word2Vec 모델을 사전 학습한다. Word2Vec 모델을 학습하는 과정에서 자주 나타나지 않는 단어를 제외하고 학습을 수행한다. 학습이 완료된 후 명사 및 고유명사별로 색인 및 벡터를 이용하여 단어 사전을 생성한다.

Padding and Unk Addition에서는 단어 사전에 특수 단어 padding과 unk 2개를 추가한다. 가변 문장의 길이를 동일하게 맞추기 위하여 padding을 추가하였다. 단어 사전에 포함되지 않는 단어를 표시하기 위해 unk를 추가하였다. padding 및 unk의 벡터는 0으로 설정하였다.

Word Dictionary Saving에서는 단어 사전을 저장한다. Word2Vec을 통해 생성된 단어 사전에 padding와 unk를 추가한 결과를 저장한다.

3. Model Training

모델 학습 과정에서는 전 처리된 학습 리스트 기반으로 딥러닝 모델을 그림 3과 같이 학습시킨다. Input Learning List에서는 전 처리된 학습 리스트를 입력한다. 학습 리스트에는 명사 및 고유명사로 구성된 문장 및 문장에 대한 기사의 장르가 포함된다.

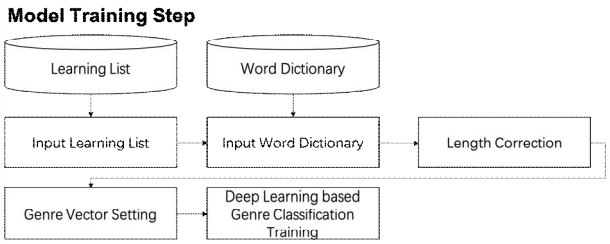

Fig. 3. Model Training

Input Word Dictionary에서는 학습 리스트에 포함된 문장의 단어들을 워드 임베딩 벡터로 변경하기 위해 단어 사전을 입력한다.

Length Correction에서는 가변 길이의 입력 문장을 고정 길이로 맞춘다. 고정된 길이보다 입력 문장이 짧은 경우 padding으로 채우고, 문장이 긴 경우에는 초과한 나머지 단어들을 삭제한다.

Genre Vector Setting에서는 기사의 장르를 기반으로 각각 문장에 장르를 부여한다. 장르 벡터는 One-hot-encoding로 해당 장르의 색인을 1로 구성하였다.

Deep Learning based Genre Classification Learning에서는 문장 리스트와 장르 벡터를 이용하여 세 가지의 딥러닝 모델을 통해 분류 학습을 진행한다. 딥러닝 모델에서 분류한 결과와 실제 장르 벡터를 비교하여 정확도 및 손실을 계산한다. 손실이 최소화되도록 딥러닝 모델을 갱신해가며 학습을 진행한다.

첫 번째, LSTM 모델은 그림 4와 같이 Embedding 층, LSTM 층, Softmax 층으로 구성된다. Embedding 층에서 길이가 보정된 문장 리스트는 Word2Vec으로 사전 학습한 문장 벡터를 워드 임베딩 하여 LSTM 층으로 입력한다. 단어 사전에 없는 단어들은 unk의 벡터로 표현된다. LSTM 층에서 문장 벡터는 LSTM 층의 가중치들과 연산해서 출력을 계산한다. Softmax 층은 LSTM 층의 출력을 장르 벡터의 차원과 동일한 확률 분포 벡터로 반환한다. 확률 분포 벡터에서 최대 확률 값을 Argmax를 통해 계산해서 기사의 장르 분류 결과를 예측한다. cross entropy와 Adam를 사용하여 최적화된 학습모델을 구축하기 위해 LSTM 층의 가중치를 갱신하며 학습을 진행한다.

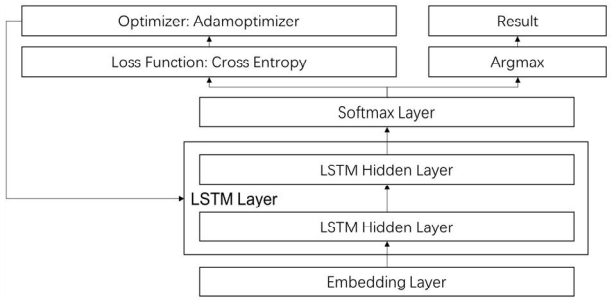

Fig. 4. LSTM Model

두 번째, GRU 모델은 그림 5와 같이 LSTM 모델과 동일하게 학습하며 LSTM 층 대신 GRU 층을 사용하여 학습한다.

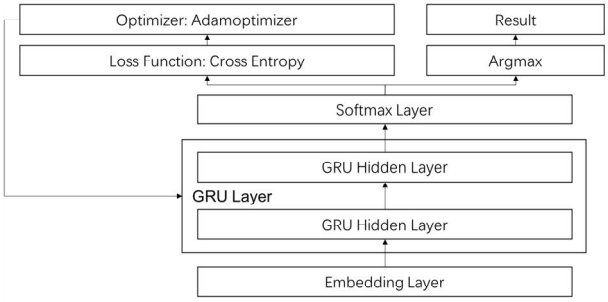

Fig. 5. GRU Model

세 번째, CNN 모델은 그림 6과 같이 Embedding 층은 LSTM의 Embedding 층과 동일한 방법으로 입력 문장 벡터를 워드 임베딩하여 CNN 층에 입력한다. Convolution 층에서 문장 벡터를 필터링을 통해 추출하여 특징 벡터로 삼는다. Max Pooling 층에서는 각 특징 벡터의 최대값으로 구성된 특징 벡터 한 개를 출력한다. Softmax 층은 CNN 층의 출력을 이용하여 LSTM 모델의 Softmax 층과 동일한 방법으로 기사의 장르 분류 결과 예측 및 CNN 층을 학습한다.

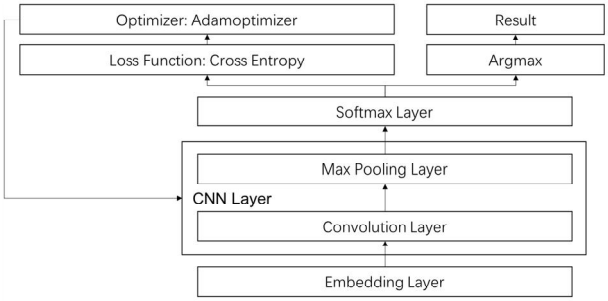

Fig. 6. CNN Model

4. The Preprocessing of Genre Selection

장르 선택용 전 처리 과정에서는 장르를 예측하기 위해서 그림 7와 같이 처리한다. 장르 선택은 기사 단위 또는 문단 단위로 입력을 받아 장르를 예측한다. Input Sentence에서는 기사 단위로 장르를 예측하는 경우에는 기사 기반을 둔 문장을 입력한다. 문단 단위로 장르를 예측하는 경우는 문단 기반을 둔 문장을 입력한다.

모델 학습용 전 처리 과정과 동일하게 문장에서 명사 및 고유명사를 추출하여 각 단어를 소문자로 변환하고 장르 테스트 문장으로 저장된다.

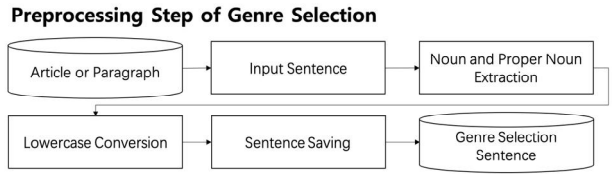

Fig. 7. Preprocessing of Genre Selection

5. Genre Selection

장르 선택 과정에서는 전 처리된 문장을 학습된 딥러닝 모델

에 입력하여 그림 8과 같이 장르를 선택한다. Input Genre Selection Sentence에서는 전 처리된 장르 테스트 문장을 입력한다. Input Word Dictionary에서는 모델 학습의 전 처리 과정에서 생성된 단어 사전을 입력한다.

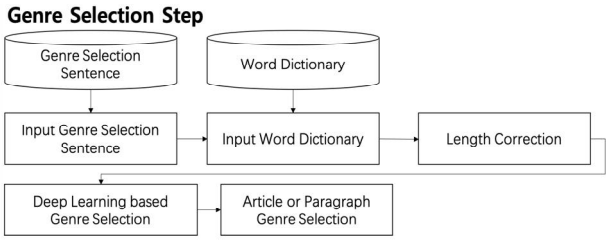

Fig. 8. Genre Selection

Length Correction에서는 장르 테스트 문장의 길이를 딥러닝 학습 모델의 문장 길이와 동일하게 변경한다.

Deep Learning based Genre Selection에서는 테스트 입력 문장을 단어 사전을 통해 LSTM, GRU, CNN의 딥러닝 학습모델에 워드 임베딩하여 장르 벡터를 예측한다. 장르 벡터는 확률 분포 벡터 중 가장 확률이 큰 값을 가지는 인덱스로 구성한다.

Article or Paragraph Genre Selection에서는 기사를 기준으로 장르 벡터를 선택하는 경우 딥러닝 모델에서 출력을 기사의 장르로 선택한다. 문단 단위로 장르 벡터를 선택하는 경우 딥러닝 모델에서 출력을 문단의 장르로 선택한다.

IV. Experiment

4장에서는 영어 기사 또는 문단을 딥러닝 모델에 입력하여 장르를 분류한 결과를 분석한다. 영어 기사 또는 문단 단위별 장르 분류 시스템 구현을 위해 구축한 실험 환경의 데이터셋, 하드웨어 그리고 소프트웨어를 기술한다.

1. Data

실험에서는 기사 또는 문단을 사용하여 장르를 분류하기 위해서 영어 말뭉치 COCA를 사용하였다[1]. COCA는 미국 영어 텍스트로 구성된 대용량의 말뭉치이며, 다섯 가지 장르인 Academic, Fiction, Magazine, Newspaper, Spoken로 구분된다. 1990부터 2017년까지 매년마다 각 장르별로 수집된 텍스트를 제공된다.

본 실험에서는 1990년부터 2012년까지의 텍스트 데이터를 사용했으며, 네 가지 장르 Academic, Fiction, Magazine, Newspaper를 사용하였다. 사용한 네 가지 장르와 달리 Spoken 장르는 대화체 텍스트이기 때문에 사용하지 않았다.

1990년부터 2012년까지의 각 장르에 포함된 article 및 paragraph의 개수는 표1과 같다.

표 2과 같이 각 기사의 시작은 “##”과 기사의 고유숫자인 일곱 자리의 숫자로 헤더가 구성되며, 문단은 기사 내에서 태그 <p>로 구분된다.

Table 1. Number of Articles and Paragraphs in COCA

Genre	Article	Paragraph
Academic	21,226	891,400
Fiction	19,276	915,000
Magazine	53,198	654,600
Newspaper	57,037	2,471,500

Table 2. COCA Database Sample

```
##4000161 Section : Life and Letters <p> I don't remember…
##4000162 Section : Life and Letters <p> The prince of wales…
##4000163 Section : Life and Letters <p> In the house of…
……
##4120083 Purpose : To review the contribution of recent…
##4120084 Purpose : In this study, the authors used cued…
##4120085 Purpose : In this study, the authors investigated…
```

본 실험에서는 COCA 말뭉치를 랜덤하게 섞어서 장르를 균등하게 분할하여 분류 학습용 데이터와 장르 예측용 테스트 데이터로 구성하였다. 기사 기준으로 장르 분류에 사용할 때에는 전체 데이터 중 최소 기사 개수를 갖고 있는 장르 Fiction의 총 기사 개수와 동일하게 모든 장르의 기사 개수를 맞춘 뒤 90%를 장르 분류 학습용 데이터로 남은 10%를 장르 선택용 테스트 데이터로 구성하였다.

문단 기준으로 장르 분류에 사용할 때에는 장르 Magazine의 총 문단의 개수를 모든 장르의 문단의 개수를 맞춘 뒤 90%를 장르 분류 학습용 데이터로 남은 10%를 장르 선택용 데이터로 구성하였다.

2. Hardware and Software Configuration

실험에서 사용한 하드웨어는 표 3과 같다. 딥러닝 모델 학습 시간을 단축하기 위해 사용한 GPU의 사양은 NVIDA GTX 1080을 사용하였다. CPU는 i5-2500K이며, Memory는 32G로 구성하였다. HDD는 256G SSD를 사용하였다.

Table 3. Hardware Configuration

Name	Version
GPU	NVIDA GTX 1080
CPU	i5-2500K
Memory	32G
HDD	256G SSD

실험에서 사용한 소프트웨어는 표 4와 같다. python 기반 프로젝트 진행을 위해 python 개발 툴킷인 PyCharm을 이용하였다. 자연어 처리 툴킷 (Natural Language Toolkit)인 NLTK 라이브러리를 이용하여 문장의 구조를 분석하였다. Word2Vec 모듈은 Gensim를 이용하였다. TensorFlow를 이용하여 딥러

닝 모델을 구현하였다. 단어 간의 맥락적 의미 관계를 시각적으로 표현하기 위해 T-SNE툴을 사용하였다.

Table 4. Software Configuration

Name	Version	Role	Class
PyCharm	2018.1.2	Programming integrated development environment	Integrated development environment
NLTK	3.2.5	Natural language processing library	Library
Gensim	3.4.0	Open source for word expression	Open source
TensorFlow	1.10.0	Deep learning open source	Open source
T-SNE	0.19.2	Data visualization	Open source

3. The Preprocessing of Model Training

본 절에서는 기사 또는 문단 단위로 장르를 분류하는 학습 모델 구축 단계에서 수행한 전 처리과정 결과를 기술한다.

Article List and Article Tagging List Generation에서는 COCA 말뭉치에 있는 기사를 기호 "##"를 기준으로 추출하였으며, "Section", "INTRODUCTION", "ABSTRACT" 등 과 같은 특정 단어들을 기사에서 제거하였다.

표 5와 같이 추출한 기사들을 모은 기사 리스트와 각각 기사에 장르를 부여한 태깅 리스트를 생성하였다.

Table 5. Article and Article Tagging List

Article List
[Life and Letters <p> I don't remember hearing …]
[Life and Letters <p> The prince of wales, in his …]
[Life and Letters <p> In the house of literature …]
…
[To review the contribution of recent studies on …]
[In this study, the authors used cued shadowing …]
[In this study, the authors investigated sentence…]
Article Tagging List
[academic]
[academic]
[academic]
…
[academic]
[academic]
[academic]

문단에 대해서 장르를 분류하는 경우 Paragraph List and Paragraph Tagging List Generation 에서 문단 표시 기호 "<p>"를 기준으로 문단을 추출하였다. 표 6의 예시처럼, 추출한 문단 리스트와 각각 문단에는 해당 기사의 태깅 정보를 이용하여 장르를 부여하였다. 예를 들어, 기사를 여러 개의 문단으로 분할했을 경우 "Life and Letters"와 같이 세 단어로 구성된 문단을 찾을 수 있었다.

Sentence List and Sentence Tagging List Setting에서는 입력 데이터가 기사인 경우 문장 리스트 및 태깅 리스트를 기사 리스트 및 기사 태깅 리스트로 설정했다. 입력 데이터가 문단인 경우는 문장 리스트 및 태깅 리스트를 문단 리스트 및 문단 태깅 리스트로 설정하였다.

Table 6. Paragraph and Paragraph Tagging List

Paragraph List
[Life and Letters]
[I don't remember hearing … was something I had to absorb.]
[In the 1960s, particularly the … to the urgencies of repentance.]
[I think what made the difference … that would bring redemption.]
…
Paragraph Tagging List
[academic]
[academic]
[academic]
[academic]
…

Noun and Proper Noun Extraction에서는 nltk 라이브러리의 형태소 분석함수 pos_tag를 이용하여 품사가 NN, NNS, NNP, 그리고 NNPS인 단어들을 추출하여 명사 및 고유명사를 기반으로 문장 리스트를 생성하였다.

문단 기준으로 전 처리가 완료된 문장 리스트와 태깅 리스트는 표 7의 예시와 같다.

Table 7. Sentence and Tagging List

Paragraph Standard Sentence List
[life, letters]
[phrase, guilt, impression, guilt, blacks, …]
[absorption, cowardice, lines, power, plates, earth, …]
[difference, fifties, sixties, guilt, fail, turmoil, rights, …]
…
Paragraph Standard Tagging List
[academic]
[academic]
[academic]
[academic]
…

Learning List Saving에서는 문장 리스트 및 장르 태깅 리스트를 통합해서 기사와 문단 단위로 학습 리스트를 저장하였다. 문장 길이는 문장에 포함된 단어의 개수를 의미하며 표 8과 같다. 최대 문장 길이가 33,199인 문단을 한 개 갖고 있는 기사도 찾을 수가 있었다. 또한, 문단에 포함된 단어 개수가 적은 경우에 장르를 분류하는데 어려움이 있었다.

기사의 최소 문장 길이는 6이며, 최대 문장 길이는 33,199이다. 본 실험에서는 딥러닝 모델의 입력 데이터가 기사인 경우 문장 길이는 실험을 통해서 실제 기사에 포함된 단어 개수보다 적게 설정하였다.

Word Dictionary Generation에서는 학습 리스트에서 자주 출현하지 않는 단어를 빈도수 기준으로 최소 3개로 설정해서 제거했으며, 10회에 걸쳐서 Word2Vec 모델의 사전 학습을 통해 단어 사전을 생성하였다. 각 단어는 300개의 실수 벡터로 lookup 테이블을 구성하도록 설정하였다. Padding and Unk Addition에서는 단어 사전에 padding과 unk을 추가했으며 해당 벡터는 0으로 설정하였다.

Table 8. Maximum and Minimum Sequence Length in Article- and Paragraph-based Sentence List

	Genre	Minimum Sequence length	Maximum Sequence length
Article	Academic	24	20,893
	Fiction	6	33,199
	Magazine	17	25,896
	Newspaper	16	23,854
Paragraph	Academic	1	12,030
	Fiction	1	33,199
	Magazine	1	21,403
	Newspaper	1	5,104

Word Dictionary Saving에서 저장한 단어 사전의 구성은 표 9과 같다. 단어 사전은 총 45,037개의 단어로 구성되며, 각 단어의 표상은 300 차원의 연속벡터로 설정되었다. 하지만 One-hot-encoding로 단어 벡터를 표상할 경우 45,037차원의 이산 벡터가 생성된다.

Table 9. Preprocessing of Model Training

Index	Word	Vector
0	pad	[0, 0, …,0, 0]
1	unk	[0, 0, …,0, 0]
2	time	[-0.5823, 0.4852, …, -0.6662,-0.3512]
…		
21,315	melton	[-0.0283, -0.0573, …, -0.0239, -0.0290]
21,316	renteria	[-0.0052, -0.0436, …,-0.01754, -0.0162]
21,317	armour	[-0.0144, -0.0296, …,-0.0556, -0.0415]
…		

각 장르 별 단어 간의 문맥 정보를 확인해보기 위해 사전 학습된 Word2Vec의 워드 벡터를 이용하여 T-SNE 툴을 통해 시각화한 결과는 그림 9와 같다. 자주 같이 나타난 단어들은 두 단어 사이의 거리가 가깝게 나타났지만 그렇지 않은 단어들은 두 단어 사이의 거리가 멀게 나타났다. 그림 9(a)의 Academic 장르에 포함된 단어들 중 university, college 등 교육과 과학의 관련 단어들이 가깝게 나타났다. 그림 9(b)의 Fiction 장르에는 사람과 관련된 단어 man, body 등이 나타났다. 그림 9(c)의 Magazine 장르 경우 president, money 및 company 등 정치와 금융에 관련된 단어들이 포함되었다. 그림 9(d)의 Newspaper 장르 경우 시간과 정부의 관련 단어가 포함되었다.

4. Deep Learning Model-based Genre Classification Training

본 절에서는 COCA 말뭉치에 대한 장르 분류 학습 모델 구축 단계에서 수행한 딥러닝 모델별 학습과정 결과를 기술한다.

Input Learning List에서는 전 처리된 문장 리스트를 입력하였다. 문장 리스트와 태깅 리스트 중 임의의 10%를 검증용으로 사용하였다.

Length Correction에서는 가변 문장 길이를 동일하게 변경하기 위해 TensorFlow 내의 함수 pad_sequences를 사용하였다.

Genre Vector Setting에서는 분류할 장르의 개수가 네 개이므로 장르벡터를 4차원 벡터로 설정하였다. 태깅 리스트의 각 태깅은 One-hot-encoding 방식으로 Academic 장르는 [1, 0, 0, 0], Fiction 장르는 [0, 1, 0, 0], Magazine장르는 [0, 0, 1, 0], Newspaper장르는 [0, 0, 0, 1]로 설정되었다.

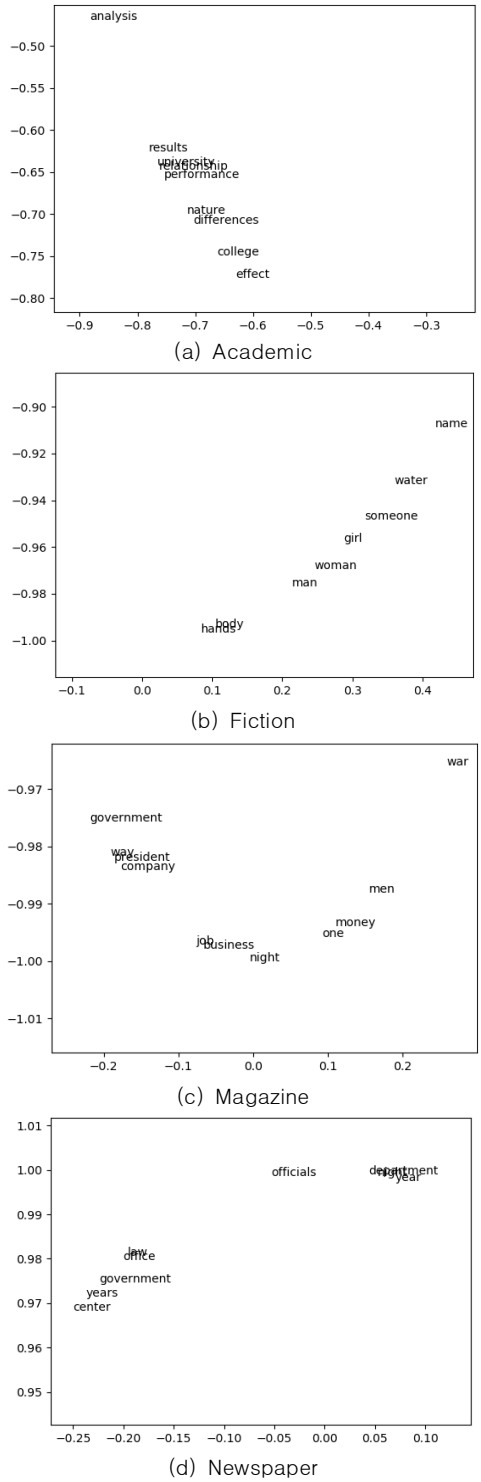

Fig. 9. Word2Vec Visualization using T-SNE

Deep Learning based Genre Classification Training에서 LSTM 모델과 GRU 모델 그리고 CNN 모델이 사용한 파라미터들은 표 10-11과 같다. 네 가지 장르를 분류하기 때문에 Num Classes 파라미터는 4로 설정하였다. Num Layers, Hidden Dim, Dropout Keep Probability, Learning Rate, Batch Size 파라미터는 LSTM 모델과 GRU 모델에서 동일하게 설정하였다.

본 실험에서는 최적의 문장 길이를 결정하기 위해 문장 길이를 달리 설정하면서 LSTM 모델과 GRU 모델의 학습 결과와 장르 예측 결과를 분석하였다.

Table 10. Parameters of LSTM and GRU

Parameter	LSTM Value	GRU Value
Num Classes	4	4
Num Layers	2	2
Hidden Dim	128	128
Drop Keep Probability	0.8	0.8
Learning Rate	0.001	0.001
Batch Size	256	256

Table 11. Parameters of CNN

Parameter	Value
Num Classes	4
Num Filters	256
Kernel Size	5
Drop Keep Probability	0.5
Learning Rate	0.001
Batch Size	256

기사에 대한 장르를 분류하기 위해 입력 문장 길이를 60으로 설정했을 때, LSTM 모델, GRU 모델, CNN 모델이 각각 50 epoch 학습하는 동안 각 epoch마다 학습 데이터의 정확도와 검증 데이터의 정확도는 그림 10과 같다. LSTM 모델, GRU 모델, CNN 모델은 학습 초기의 정확도는 70%로 유사하였다. 하지만, CNN 모델은 epoch 11일 때 학습 정확도가 약 99%에 도달하였다. LSTM 모델 및 GRU 모델은 epoch 21일 때 학습 정확도가 약 99%에 도달하였다.

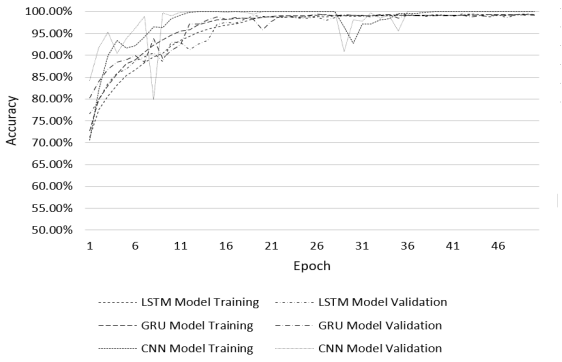

Fig. 10. Deep Learning Model-based Training and Validation Accuracy in Classifying Articles

기사의 장르 분류에 대한 학습 데이터와 검증 데이터의 손실 변화는 그림 11과 같다. CNN모델이 다른 모델들보다 epoch이 1~5 진행하는 동안 가장 급격하게 감소하였다. CNN 모델은 epoch 11 부터 손실이 약 0.009에 근접하였지만, LSTM 모델과 GRU 모델은 약 0.02에 수렴하였다. GRU 모델의 손실이 LSTM 모델보다 빨리 수렴하였다.

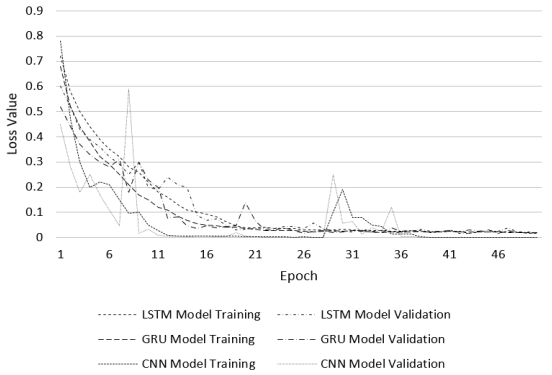

Fig. 11. Loss Change In Classifying Articles,

그림 12에서처럼 문단의 장르를 분류하기 위해 문장 길이를 41로 설정하고, 각 모델을 epoch 50으로 학습 진행했을 때 LSTM 모델과 GRU 모델은 학습 데이터의 정확도가 약 85%이였고 CNN 모델은 약 99%의 학습 데이터의 정확도를 보였다.

문단의 장르 분류에 대한 손실 변화는 그림 13과 같다. CNN 모델의 손실은 0에 수렴하였으나, epoch이 진행하는 동안 검증 데이터의 손실 변화가 큰 것을 확인하였다. LSTM 모델과 GRU 모델은 손실이 약 0.3에 수렴하였다.

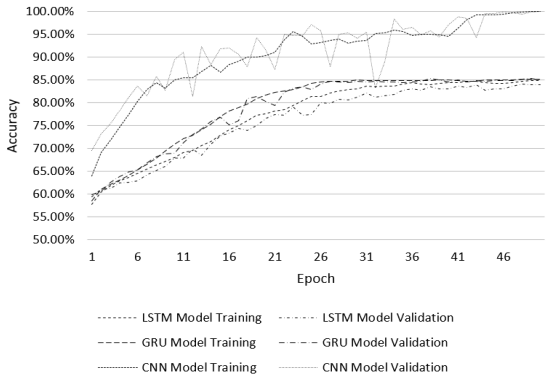

Fig. 12. Model Training Accuracy and Validation Accuracy Change In Classifying paragraphs,

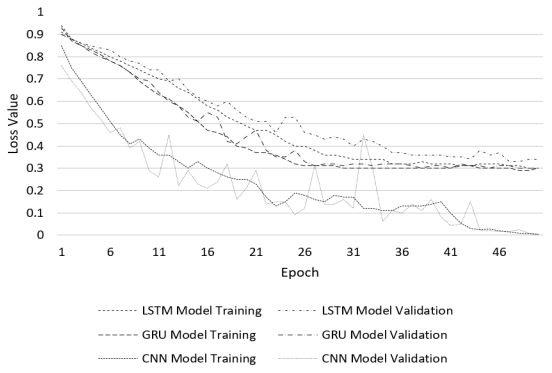

Fig. 13. Loss Change In Classifying Paragraphs,

5. Preprocessing for Genre Selection in Genre Prediction

본 절에서는 기사 또는 문단을 선택해서 학습된 모델을 이용하여 해당 장르를 예측하는 분류 시스템을 구축하는 단계에서 수행한 테스트 데이터에 대한 전 처리 과정 결과를 기술한다.

Input Sentence에서는 COCA 말뭉치에서 학습할 때 사용하지 않고 남겨둔 10%의 데이터를 테스트 문장으로 입력하였다.

테스트 기사에 대한 장르 문장을 선택하는 경우에는 입력 문장을 기사로 입력했으며, 테스트 문단에 대한 장르 문장을 선택하는 경우는 입력 문장을 문단으로 입력하였다. 나머지 전 처리 과정은 분류 학습 단계의 모델 학습용 전 처리와 동일하다.

6. Deep Learning Model-based Genre Selection in Genre Prediction

본 절에서는 학습된 세 개 딥러닝 모델을 이용하여 기사 또는 문단을 선택 입력하여 해당 장르를 예측하는 시스템을 구축하는 단계에서 수행한 장르 예측 결과를 기술한다.

Input Genre Selection Sentence에서는 전 처리한 장르 선택 문장을 입력하였다. Input Word Dictionary에서는 Word2Vec의 단어 사전을 입력하였다. Length Correction에서는 학습 과정에서 지정한 문장 길이를 기준으로 장르 선택 문장의 길이를 보정하였다.

Deep Learning based Genre Selection에서는 장르 선택 문장을 학습된 딥러닝 모델에 입력하여 [0, 0, 0, 1]과 같은 4 차원의 장르 벡터가 출력되었다. Article or Paragraph Genre Selection에서는 기사 또는 문단에 대한 장르 예측하는 시스템의 출력 결과를 장르 벡터에서 문자열 장르로 변환하였다.

제안한 시스템의 실험 결과는 기사 장르 분류 학습에서 문장 길이를 COCA 말뭉치에 있는 기사의 최대 길이로 설정하여 학습하는 경우 하드웨어의 메모리 부족 문제가 발생하였다. 또한 기사 및 문단에 포함된 단어들을 모두 딥러닝 모델에 학습하는 경우 단어 사전의 크기가 충분히 크지 않아서 학습 성능이 저조하였다.

따라서 위의 문제를 해결하기 위해 첫 번째, 기사 단위로 장르를 분류 학습하는 과정에서 문장 길이를 달리하였을 때 장르 분류에 미치는 영향을 확인하기 위해서 문장 길이를 30, 40, 50, 60으로 다르게 설정하여 실험 결과를 분석하였다.

학습모델을 epoch 50으로 수행했을 때 학습 및 검증의 정확도, 장르 예측의 정확도 결과는 표 12와 같다. 모델 학습 및 검증의 정확도는 CNN 모델이 가장 높았다. 하지만, 장르 예측의 정확도는 LSTM 모델이 가장 높았으며, 다음으로는 GRU 모델의 정확도가 높았다. LSTM 모델, GRU 모델, CNN 모델 모두 문장 길이가 길어지면 정확도가 향상 되는 것을 확인하였다.

Table 12. Training Accuracy, Validation Accuracy, and Genre Test Accuracy in the Articles Experiment

Sequence length	Model	Model Training Accuracy	Model Validation Accuracy	Genre Test Accuracy
30	LSTM	99.49%	99.56%	**74.18%**
	GRU	99.38%	99.58%	73.43%
	CNN	**99.94%**	**99.96%**	71.11%
40	LSTM	**99.44%**	**99.74%**	74.47%
	GRU	99.43%	99.61%	**74.90%**
	CNN	98.26%	99.67%	73.11%
50	LSTM	99.33%	99.38%	**77.43%**
	GRU	99.20%	99.56%	76.19%
	CNN	**99.97%**	**99.99%**	75.01%
60	LSTM	99.37%	99.31%	**78.28%**
	GRU	99.30%	99.22%	78.24%
	CNN	**99.97%**	**99.97%**	76.12%

두 번째, 문단 단위로 장르를 분류 학습하는 과정에서 문단을 구성하는 단어 개수를 구간으로 나누어 각 구간별 문단을 분류하는데 미치는 영향을 분석하였다. 실험에서는 단어 개수를 6개 구간 (2-10, 11-20, 21-30, 31-40, 41-50, 51-60)으로 나누고, 각 구간에 해당하는 단어들을 추출하여 장르를 분류하는 실험을 진행하였다. 길이가 1인 문단을 분류하는 것은 의미가 없기 때문에 실험하기 전에 제거하였다. 문단의 단어 개수가 30이하인 경우에는 장르와 연관된 단어가 매우 적게 나타났다. 또한 60개 이상 단어를 포함하는 문단들은 학습할 만큼 충분하지 않았다. 따라서 단어의 개수가 31-40, 41-50, 51-60개로 구성된 3 개의 구간을 대상으로 학습한 결과는 표 13과 같다. LSTM 모델 및 GRU 모델의 장르 예측 정확도가 CNN 모델의 정확도보다 더 높았다. 41-50개의 단어로 구성된 문단의 장르 예측 결과가 31-40개의 단어로 구성된 문단보다 더 높게 나타났다. 문단에 포함된 단어가 적어도 41~50개인 경우 67.00%로 장르 분류가 가능하였다.

V. Conclusions

대용량의 현대 영어 코퍼스를 대상으로 Word2Vec, LSTM, GRU, CNN 딥러닝 기술을 활용하여 기사 또는 문단 단위로 임베딩 벡터를 사용하여 장르를 분류하는 시스템을 설계 및 구현하였다.

64,800개의 기사를 장르 4개로 분류하는 모델로 학습하고 7,200개의 기사에 대한 장르를 예측하였을 때, 입력 문장 길이가 60인 경우 분류 정확도는 LSTM, GRU, CNN 순서로 나타났다. 또한, 1,800,000개의 문단을 장르 4개로 분류하는 모델로 학습하고 200,000개의 문단에 대한 장르를 예측했을 때, 문장 길이가 41인 경우 분류 정확도는 LSTM, GRU, CNN 순서로 나타났다.

Table 13. Training Accuracy, Validation Accuracy and Genre Test Accuracy in the Paragraphs Experiment

Sequence length (Sequence length Section)	Model	Model Training Accuracy	Model Validation Accuracy	Genre Test Accuracy
31 (31-40)	LSTM	82.99%	82.22%	**64.56%**
	GRU	83.24%	83.49%	64.34%
	CNN	**97.80%**	**96.45%**	61.11%
41 (41-50)	LSTM	84.92%	83.92%	**67.00%**
	GRU	85.09%	85.08%	64.96%
	CNN	**99.99%**	**99.98%**	64.02%
51 (51-60)	LSTM	74.60%	75.30%	**64.30%**
	GRU	74.93%	76.00%	63.90%
	CNN	**92.34%**	**89.35%**	62.20%

실험 결과는 다음과 같이 해석할 수 있다. (1) LSTM 모델이 GRU와 CNN모델과 성능을 비교했을 때, LSTM 모델이 더 높은 것으로 보아 텍스트 분류 문제는 전체 텍스트의 시퀀스를 학습하는 것이 텍스트의 특징을 통해 학습하는 것보다 더 올바른 접근이라고 생각할 수 있다. (2) 문단보다 기사의 장르를 분류할 때 모든 딥러닝 모델의 예측 정확도가 더 높은 것으로 보아 텍스트 분류는 짧은 문서보다 긴 문서를 통해 학습하는 것이 더 올바른 접근이라고 생각 할 수 있다. (3) 학습 데이터의 기사 입력 처리는 기사의 문단들을 순차적 처리하지만, 문단 입력 처리는 기사를 구성하는 문단들은 독립적 관계이며 동일한 장르를 갖는 경우로 간주하기 때문에 기사를 구성하는 전체 문단의 순서도 모델 학습에 반영된다고 생각할 수 있다. (4) LSTM과 GRU 모델은 학습 과정과 장르 예측 과정의 성능 결과가 비슷하게 나타났다. 하지만 CNN 모델의 경우에는 학습 과정과 장르 예측의 성능 차이가 LSTM과 GRU 모델보다 큰 것으로 보아 텍스트 분류할 때 단어 사전에 포함되지 않은 단어가 CNN 모델 학습에 영향을 준 것으로 판단할 수 있다.

REFERENCES

[1] COCA, https://corpus.byu.edu/coca/

[2] Sejong Corpus, https://ithub.korean.go.kr/user/main.do

[3] J. Swales, “Genre Analysis: English in Academic and Research Settings,” Cambridge University Press, 1990.

[4] D. Biber, “Variation across Speech and Writing,” Cambridge University Press, 1988.

[5] D. M. Blei, “Probabilistic Topic Models,” Communications of the ACM, Vol. 55, No. 4, 77-84, Apr. 2012.

[6] Z. S. Harris, “Distributional Structure,” pp.775-794, Springer, 1997.

[7] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” Machine Learning 29.2-3, pp.131-163, Nov. 1997.

[8] H. Jo, J-H. Kim, S. Yoon, K-M. Kim, and B-T. Zhang, “Large-Scale Text Classification with a Convolutional Neural Network,” 42th The Korean Institute of Information Scientists and Engineers Annual Meetings, 2015.

[9] H. Jo, J-H. Kim, K-M. Kim, J-H Chang, J-H. Eom, and B-T. Zhang, “Large-Scale Text Classification with Recurrent Neural Networks,” 43th The Korean Institute of Information Scientists and Engineers Annual Meetings, 2016.

[10] T. Young, D. Hazarika, S. Poria, E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing,” arXiv:1708.02709, Oct. 2018.

[11] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient estimation of word representations in vector space,” arXiv:1301.3781, Jan. 2013.

[12] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” International Conference on Machine Learning, pp. 1188-1196, Jan. 2014.

[13] C. Goller and A. Kuchler, “Learning task-dependent distributed representations by backpropergation through structure,” Neural Networks, IEEE International Conference, Vol. 1, 1996.

[14] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation 9.8, pp. 1735-1780, Nov. 1997.

[15] K. Cho, et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.

[16] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architecture,” Proceedings of the 32nd Intenational Conference on Machine Learning, 2015.

[17] Y. LeCun and Y. Bengio, “Convoluntional networks for images, speech, and time series,” In M. A. Arbib (Ed.), The handbook of brain theory and neural networks, Cambridge, MA: MIT Press, pp. 255-258, 1995.

[18] Yoon Kim, “Convoluntional Neural Networks for Sentence Classification”, Empirical Methods on Natural Language Proceeding, 2014.

[19] Y. Liu and M. Zhang, “Neural Network Methods for Natural Language Processing”, Computational Linguistics, Vol. 44, pp.193~195, Mar. 2018.

[20] E-S. You, G-H. Choi, and S-H. Kim, “Study on Extraction of Keywords Using TF-IDF and Text Structure of Novels”, Journal of The Korea Society of Computer and Information, Vol. 20(2), pp. 121-129, Feb. 2015.

[21] J. Park, H. Kim, H-G. Kim, T-K. Ahn, and H. Yi “Structuring of Unstructured 눈 Messages on Rail

Services using Deep Learning Techniques", Journal of The Korea Society of Computer and Information, Vol. 23(7), pp. 19-26, Jul. 2018.

Authors

Euhee Kim received the M.S. degrees in Computer Engineering from Dongguk University, Korea, in 2002 and Ph.D. degrees in Mathematics from The University of Connecticut, U.S.A. in 1995. Dr. Kim is currently an associate Professor in the Department of Computer Science & Engineering at Shinhan University. She is interested in AI and Big Data computing.