Word2vec을 활용한 RNN기반의 문서 분류에 관한 연구

Text Document Classification Based on Recurrent Neural Network Using Word2vec

저자 (Authors)

김정미, 이주홍

Jung-Mi Kim, Ju-Hong Lee

출처 (Source)

한국지능시스템학회 논문지 27(6), 2017.12, 560-565 (6 pages)

Journal of Korean Institute of Intelligent Systems 27(6), 2017.12, 560-565 (6 pages)

발행처 (Publisher)

한국지능시스템학회

Korean Institute of Intelligent Systems

URL

http://www.dbpia.co.kr/Article/NODE07284295

APA Style

김정미, 이주홍 (2017). Word2vec을 활용한 RNN기반의 문서 분류에 관한 연구. 한국지능시스템학회 논문지, 27(6), 560-565.

Journal of Korean Institute of Intelligent Systems
Vol. 27, No. 6, December 2017, pp. 560–565
http://dx.doi.org/10.5391/JKIIS.2017.27.6.560

ISSN(Print) 1976-9172
ISSN(Online) 2288-2324

JKIIS

Word2vec을 활용한 RNN기반의 문서 분류에 관한 연구

Text Document Classification Based on Recurrent Neural Network Using Word2vec

김정미 · 이주홍†
Jung-Mi Kim and Ju-Hong Lee†

인하대학교 컴퓨터 · 정보공학과
Department of Computer and Information Engineering, INHA University

요 약

자연어 처리 분야에서도 심층 신경망 기술이 주목되고 있으며, 최근에는 convolutional neural network (CNN)기반의 심층 신경망 구조가 이미지 분류뿐만 아니라 자연어 처리의 문서 분류에서도 좋은 성능이 입증되었다. 하지만 convolutional neural network (CNN)을 이용한 문서 분류 연구에서는 문장의 평균 단어 수가 16개로 이루어진 짧은 문장에 한하여 적용되었으며, 구문 전체와 의미론적 관계가 복잡한 전체 문장을 다루기 어렵다는 단점을 가지고 있다. 본 논문은 기존 연구의 한계점을 극복하고 더 정확한 문서 분류 성능을 위하여 word2vec를 활용한 recurrent neural network (RNN)기반의 심층 신경망의 접근법을 새롭게 제안한다. 이를 위해 장기 의존성 문제를 해결한 long short-term memory (LSTM)을 사용하여 긴 시퀀스의 입력에서도 효과적인 문서 분류가 가능하도록 하였고, 제안 방식의 효율성을 검증하기 위해 영문 데이터 뿐 아니라 한국어 영화 리뷰 데이터에 대해서도 실험을 수행하였다. 그 결과 장문을 포함하고 있는 영문 신문 기사에서는 87%, 단문으로 구성된 영문 영화 리뷰 문서에서는 90%, 한국어 영화 리뷰에서는 88%의 문서 분류 정확도를 보였다.

키워드 : 텍스트 마이닝, 정보검색, 딥 러닝, 문서분류

Abstract

Deep neural network based methods have obtained remarkable progress on natural language processing (NLP) task. Recently, convolutional neural network (CNN) based approaches often outperform not only in image classification, but also in document classification. However, convolutional neural network (CNN) based methods is applied only to a short sentence composed of 16 words in average, and it has a disadvantage that it is difficult to deal with a sentence having a complicated semantic relationship with the whole sentence. In this paper, we propose a new method based on recurrent neural network (RNN) using word2vec to overcome the limitations of previous related work and to get much higher accuracy of document classification. By using long short-term memory (LSTM) to solve the long-term dependency problem, effective document classification is also possible for long sequence input. To validate performance of our proposed method in various data, we tested our proposed method both with English sentence and Korean movie review dataset. As a result, 87% of the English newspaper articles containing the long texts, 90% of the English movie review and 88% of the Korean movie review showed the accuracy of document classification.

Key Words : Text Mining, Information Retrieval, Deep Learning, Document Classification

Received: Aug. 9, 2017
Revised: Oct. 10, 2017
Accepted: Oct. 10, 2017
†Corresponding authors
juhong@inha.ac.kr

1. 서 론

인터넷 기술이 발전함에 따라 많은 양의 비정형 데이터가 급격하게 증가하고 있는 빅 데이터 시대가 도래되었다. 특히 SNS에 남긴 각종 리뷰, 신문 기사, 인터넷 등의 문서 정보들이 기하 급수적으로 증가함에 따라 효율적인 정보검색을 위하여 문서를 같은 범주의 주제로 분류하는 기술의 필요성이 대두되고 있다. 최근 몇 년간의 연구에서는 자연어 처리에서도 심층 신경망(Deep Neural Networks, 이하 DNN) 기법이 좋은 성능을 나타내고 있다. 그 중에서도 컨볼루션 신경망(Convolution Neural Network,이하 CNN)이 더욱 효과적이라는 것이 알려지면서, 단어를 벡터(vector)로 표현하는 방법인 word2vec과 CNN을 이용한 문장 분류 방법이 제안되었고 실제로 우수한 결과를 보여주었다[1][2]. 그러나 CNN을 활용한 기존의 심층 신경망 방식의 문서 분류 연구에서는 대체로 짧은 영어 문장을 대상으로 한 성능 평가 실험 결과는 제시하였지만, 구문 전체와 의미론적 관계가 복잡한 문장을 다룰 때의 성능 여부는 제시하지 못했다. 또한 2017년도의 Doo wu kim[3]의 연구를 살펴보면 word2vec과 CNN기반의 제안 방식은 한국어 문서의 분류에 있어서는 성능이 좋지 않음을 알 수 있다.

이에 본 논문에서는, 기존 연구[1][2][3]의 한계점을 개선하여 긴 문장을 포함한 문서에서도 같은 범주의 주제로 문서를 정확하게 자동 분류하기 위해 word2vec을 활용한 Recurrent Neural Network(이하 RNN) 기반의 방법을 새롭게 제안한다. 심층 신경망의 효율적이고 정확한 학습을 위해서 word2vec을 사용하여 문서 내의 단어들에 대한 신뢰성 높은 특징 값을 부여하고, 이를 LSTM (Long Short Term Memory, 이하 LSTM) 의 입력 값으로 활용하여 문서 분류를 위한 변별적인 자질 값을 생성해 낸다. 또한, 다양한 구성의 데이터에서도 효율적인 문서 분류의 성능 검증을 위하여 한국어로 작성된 영화 평점 리뷰를 대상으로도 실험하여 영어 문서 뿐 아니라 한국 문서 분류에서도 본 논문의 제안 방식이 효과적임을 입증한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안 방식과 관련된 이론적인 배경, 3장에서는 word embedding을 활용한 데이터 전처리와 제안 방식의 시스템 구조를 자세히 설명한다. 4장에서는 실험 데이터 셋의 구성과 실험 데이터의 전처리 과정, 그리고 문서 분류를 위한 기존 연구와의 비교 실험을 수행하고 결과를 분석한다. 마지막으로 5장에서 본 연구의 결론 및 향후 연구 방향을 제시한다.

2. 이론적인 배경

2.1 word embedding

단어의 순서와 의미를 내포하는 벡터의 형태로 단어를 표현하는 기법으로 word2vec 모델이 가장 대표적이다. word2vec는 특정 embedding 공간상에서 같은 맥락(context)을 갖는 단어들이 가까운 거리를 가진다는 전제(Distributional Hypothesis)에서 출발한다[4][5].

이러한 word embedding 방식의 word2vec 표현법은 주어진 문장에 대한 문법적 해석이 가능하며, 단어의 거리를 통해 의미론적 추론도 가능하다는 것이다. 그림 1의 (a)는 문법적 특징에 따라 단어 벡터가 갖는 방향성을 도식화한 것이다. 많은 양의 문서 데이터를 학습할수록 반복되는 문장의 맥락을 통해 break, broken의 벡터 공간에서의 거리가 have와 had의 벡터 공간에서 거리와 같아 질 수 있으며, 이는 과거의 활용되었던 단어의 의미를 관계적 맥락에 따라 embedding 공간 상에서 다르게 학습 할 수 있다는 것을 나타낸다. 그림 1의 (b)는 의미론적인 추론에 대한 예시로 “한국”에 대한 벡터에서 “서울”에 대한 벡터를 빼고, “도쿄”에 대한 벡터를 넣는다면 “일본”이라는 결과를 도출해 낼 수 있다는 것을 표현하고 있다.

정리하자면 word2vec은 주어진 문장을 구성하는 단어들의 전후 관계를 학습하여 단어의 의미를 내포하고 있는 벡터 값으로 문서를 구성하고 있는 자질들을 수치화한다. 이것은 기존의 통계적인 방식을 활용한 연구[6][7][8]와는 다르게 별도의 유사도 계산이나 차원 축소 과정 없이 변별적인 특징을 내포하고 있는 벡터 값으로 단어를 수치화 할 수 있다는 것이다. 본 연구에서는 word2vec을 활용한 데이터 전처리를 통해 문서를 구성하고 있는 각각의 자질들에게 문서의 변별적인 특징을 잘 표현하는 벡터 값을 부여하여 LSTM의 학습 성능을 높인다.

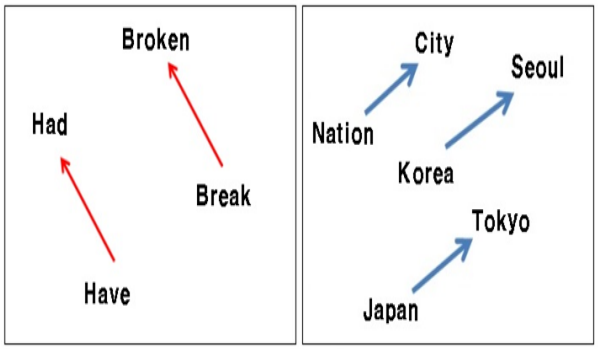

그림 1. word2vec 모델의 벡터 표현을 통한 추론 예시:
(a) 문법적 해석 (b) 의미론적 해석
Fig. 1. The example of inference by word2vec vector representation : (a)a grammatical interpretation (b)a semantic interpretation

2.2 Recurrent Neural Network (RNN)

기존의 신경망 구조에서는 입력과 출력이 각각 독립적이라고 가정하였지만, RNN에서는 동일한 활성 함수를 한 시퀀스의 모든 요소마다 적용하여 출력 결과가 이전의 계산 결과에 영향을 받는다. 하지만 RNN의 실제 구현에서는 비교적 짧은 시퀀스만 효과적으로 처리하는 한계점이 있다[9]. 이를 장기 의존성 문제(the problem of Long-Term Dependencies)라고 하며, 이 문제를 극복하고자 순환 신경망의 변형 알고리즘 LSTM이 제안 되었다. 기본 RNN의 구조에서 데이터를 계산하는 각 길목에 입력 게이트, 망각 게이트, 출력 게이트가 추가하여 각 상태 값을 메모리 공간 셀에 저장하고, 데이터를 접하는 게이트 부분을 조정하여 불필요한 연산, 오차 등을 줄여 장기 의존성 문제를 일정부분 해결하였다[10].

LSTM은 많은 자연어처리 문제에 대해서 성공적으로 적용이 되었다. 특히 주어진 문장에서 이전 단어를 보고 다음 단어가 나올 확률을 계산해주는 언어 모델이나, 자동 번역의 출력 값으로 어떤 문장을 내보내는 것이 더 좋은지 결정하는 기계 번역 분야에서 좋은 성능을 나타내고 있다. 다시 말하면, LSTM은 긴 시퀀스의 입력 값으로부터 자질의 대표적인 특징을 생성해내는 생성모델로 많이 활용되고 있으며 그 성과 또한 훌륭하다. 본 연구에서는 이러한 LSTM 특성을 활용하여 문서의 구별되는 특징을 심층적으로 훈련하고, 훈련의 출력 값으로 문서를 자동으로 분류할 수 있는 자질 값을 생성하여 자동화된 문서 분류의 성능을 높인다.

3. Word2vec을 활용한 RNN기반의 문서분류

3.1 데이터 전처리

띄어쓰기를 기준으로 문서를 토큰화하여 특수 문자가 제거된 단어로 문서를 재구성하였다. 그리고 토큰화된 단어를 벡터로 표현하기 위하여word2vec을 활용하였다[11]. word2vec을 활용하면 의미가 유사한 단어나 문법적으로 비슷한 구조를 이루는 단어는 embedding 공간 상 가까운 벡터 공간에 놓이게 된다. 즉, 문서를 표현하는 단어의 벡터 값들이 범주별로 군집화 되어 다른 범주에 속한 문서들과 벡터 공간상의 위치에 있어 구분될 수 있도록 단어의 벡터 표현이 생성된 것이다. 심층 신경망의 학습에 앞서 이러한 데이터 전처리 방식은 자질들의 변별적인 특징을 잘 내포하고 있는 벡터 값으로 단어를 수치화하여 LSTM의 학습 성능을 높일 수 있다.

3.2 시스템 구조

데이터 전처리를 통해 각 각의 자질들에게 문서의 특징을 잘 표현하는 벡터 값을 부여하였다. 이제 심층 신경망으로 이러한 자질들을 깊이 학습하여 문서를 자동으로 분류할 수 있는 자질 값을 생성한다. 그림 2은 제안 모델의 시스템 구조를 단순화 하여 표현한 것이다. LSTM의 고정 입력 길이만큼의 텍스트 시퀀스 $x = \{x_1, x_2, \cdots, x_T\}$가 주어지면 i번째 단어 x_i는 word embedding으로부터 벡터 X_i로 표현된다. h_i는 LSTM의 내부 은닉층으로 시간 단계 t에 따라 단어 벡터 표현 X_i를 순차적으로 고정 입력 길이까지 입력받는다.

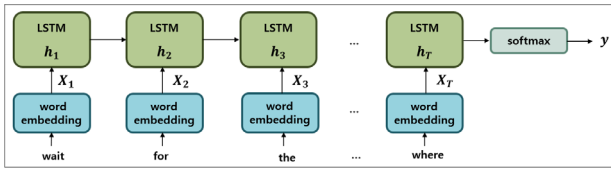

그림 2. 제안 모델 시스템 구조
Fig. 2. The structure of proposed model

h_t의 활성화는 현재 입력 x_t와 이전 은닉층 상태의 h_{t-1}의 활성함수 f로 계산이 되며 (수식 1), 마지막 순간의 h_T 출력은 시퀀스에 대한 전체 표현이 된다(수식 2).

$$h_t = \begin{cases} 0 & t = 0 \\ f(h_{t-1}, X_t) & otherwise \end{cases} \quad (1)$$

$$h_t = LSTM(\hat{X}) \quad (2)$$

fully connected층은 은닉층이 없이 분류 범주에 대한 확률 분포를 예측하는 비선형의 softmax층으로 연결되어 있다. k번째 시퀀스 입력 x_k가 입력 될 때, 예측 값 $\hat{y}_k$을 계산하기 위한 softmax 층의 계산은 (수식 3)를 따른다.

$$\hat{y}_k = \frac{e^{x_k}}{\sum_{i=1}^{n} e^{x_i}} \quad (3)$$

n은 마지막 출력 층의 뉴런 수를 나타내며, 분모는 입력된 전체 시퀀스 벡터의 지수 함수, 분자는 입력된 시퀀스 벡터 x_k의 지수함수이다. 훈련을 위한 비용함수는 cross-entropy방법을 사용하였다. 예측 값 $\hat{y}$에 로그를 취한 것과 실제 값 y의 곱을 전부 합하여 범주의 개수 m만큼 나눈 값으로, 목표는 예측 값 $\hat{y}$와 실제 값 y의 확률 분포차이를 구하는 식(수식 4)이다.

$$L(\hat{y}, y) = -\frac{1}{m}\sum_{i=1}^{m} y\log(\hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}(y-\hat{y})x \quad (4)$$

데이터의 분산을 기반으로 한 아주 작은 값으로 가중치(Weight)를 초기화를 시켰고, 편차(bias)는 0의 값으로 초기화 시켰다. LSTM을 훈련 할 때의 학습 계수는 0.001이며, 학습 알고리즘은 경사하강의 방식(gradient descent)중 하나인 adam을 사용하였다[12].

word2vec을 훈련할 때 CBOW 구조[5]를 사용하였고, 영문으로 작성된 문서에 한에서는 Google News로부터 1,000억 단어에 대해 사전 훈련된 word2vec 벡터를 사용하였다[13]. 단어의 벡터는 300차원이며 사전 훈련 된 단어가 없는 경우에는 무작위로 초기화 시켰다.

4. 실험 및 관련 연구와의 성능 비교

4.1 실험 데이터 셋 구성

제안 모델의 성능을 입증하기 위하여, 문서 분류의 목적이 다른 4가지 종류의 데이터 셋을 사용하였으며, 훈련 데이터와 테스트 데이터의 개수 및 문서 구성 형태는 표1에 자세히 기술 되어 있다.

1) 20 news data : 영문으로 작성된 20,000개의 신문 기사 데이터이며 주제에 따라 20개의 다른 범주로 분류된다. 훈련되는 문서의 양과 목표 범주의 수가 2배씩 늘어나도 모델 성능의 강건함을 입증하기 위하여 3개, 6개 및 12개의 서로 다른 뉴스 주제로 구성된 3개의 뉴스 그룹으로 나누었다. 각 뉴스 그룹의 주제는 다음과 같이 나열되며, 여기서의 NG는 News Group의 약자이다.[14]

- 3NG : comp.sys.ibm , pc.hardware, rec.sport.baseball, soc.religion.christian
- 6NG : alt.atheism, comp.sys.mac.hardware, rec.motorcycles, rec.sport.hockey, soc.religion.christian, talk.religion.misc.

• 12NG : comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, mic.forsale, rec.autos, rec.sport.baseball, sci.space, talk.politics.misc, sci.crypt, sci.med, talk.religion.misc, alt.atheism, soc.religion.christian.

2) Naver movie : 짧은 문장단위로 구성된 영화 평점 리뷰다. 한국어로 작성되었으며, 네이버 영화 네티즌 평점으로부터 수집된 실제 데이터이다. 범주는 긍정 및 부정(negative,positive)의 이원 범주를 가지고 있다.[15]

3) SST-1 : 영문 영화 평점 리뷰이며, 5가지의 범주(negative, somewhat negative, neutal, somewhat positive, positive)를 가지고 있다. 리뷰 데이터이므로, 데이터 형태는 문장이다. [16]

4) SST-2: 영문 영화 평점 리뷰이며, 긍정 및 부정(negative,positive)의 이원 범주를 가지고 있다. [16]

표 1. 본 논문에서 사용된 데이터 셋 현황
Table 1. The state of datasets used in this paper

Data set	#train	#test	#class	classification Task	Type
3NG	1487	1487	3	English news categorization	Document
6NG	2619	2619	6	English news categorization	Document
12NG	5288	5288	12	English news categorization	Document
naver movie	150000	50000	2	Korean Sentiment analysis	Sentence
SST-1	8892	2963	5	English Sentiment analysis	Sentence
SST-2	7210	2403	2	English Sentiment analysis	Sentence

4.2 LSTM의 입력 토큰 개수 선정

학습 과정에서의 메모리와 연산의 최적화를 위해 LSTM의 입력 토큰 수를 선정하는 실험을 진행한다. 그림 3는 각각의 실험 데이터에 대해서 한 문서 당 포함하고 있는 단어 수의 현황이다. 여기서의 단어는 문서의 토큰화를 진행한 후 재구성된 단어이므로 단어라는 표현을 토큰이라고 대체한다. x축은 한 문서가 포함하고 있는 토큰의 개수이고, y축은 동일한 토큰 단어 개수를 가진 문서의 출현 빈도수이다. 가장 적은 토큰 수를 가진 문서와 가장 많은 토큰 수를 가진 문서의 차이 값이 매우 크다. 때문에 한 번의 시퀀스 입력 값으로 한 문서의 모든 토큰을 활용하기 위해서 데이터 셋 내에 가장 많은 토큰 수를 가진 문서를 기준으로 입력 토큰수를 결정하고, 해당 문서보다 적은 토큰 수를 가진 문서는 0의 값을 채우는 zero-padding 방식을 활용한다면 0으로 채워야 하는 값이 늘어남에 따라 자원 사용량의 낭비 및 훈련시간 증가와 같은 비효율 문제가 야기된다. 때문에 식별 성능을 저하시키지 않는 범위 내에서 LSTM의 입력 토큰 수를 제한한다.

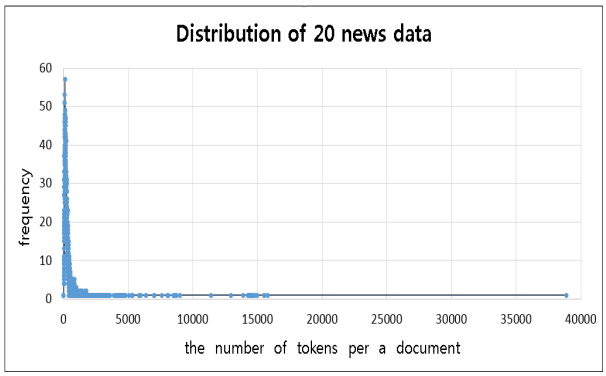

그림 3. 전체 20news data에서 동일 토큰 수 대비 문서 빈도수 현황, x축 : 문서가 포함하는 토큰의 수, y축 : 동일한 토큰 수를 가진 문서의 출현 빈도수
Fig. 3. Distribution of document frequency in whole 20 news data according to the number of token included each document, x-axis: the number of tokens the document, y-axis: the number of frequency with the same tokens in document

제한 된토큰수는 한 문서의 주제 및 내용을 판별할 수 있을 정도로 충분해야 한다. 그림 3의 그래프를 보면 대부분의 문서는 1000개의 토큰조차 포함하고 있지 않으며, 동일 토큰 수로 출현빈도가 가장 많았던 문서의 토큰 개수는 132로 전체 문서 길이에 비하면 매우 짧은 문서 길이이다. 때문에, 조금 더 자세한 현황을 파악하기 위하여 토큰의 수가 1000개 미만일 경우를 기준으로 차트를 다시 생성하였고 그림 4와 같다.

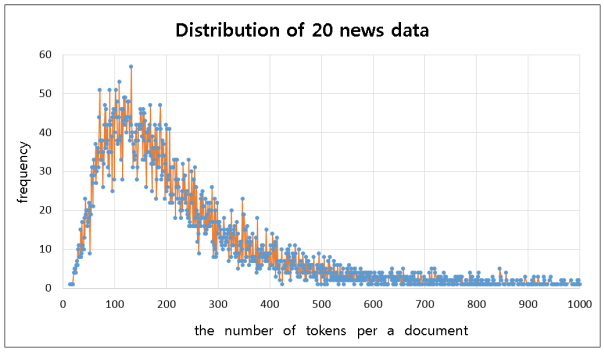

그림 4. 1000토큰 이하의 20news data에서 동일 토큰 수 대비 문서 빈도수 현황, x축 : 문서가 포함하는 토큰의 수, y축 : 동일한 토큰 수를 가진 문서의 출현 빈도수
Fig. 4. Distribution of document frequency in 20 news data less than 1000 tokens, x-axis: the number of tokens the document, y-axis: the number of frequency with the same

위의 두 그래프는 75와 200사이의 토큰 수만으로도 대부분의 문서에서 충분히 내용과 주제를 판별할 수 있다는 것을 의미한다. 때문에 동일 토큰 수 대비 문서 출현 빈도를 내림차순 정렬하여 상위 50의 평균, 상위 30의 평균, 상위 10의 평균, 가장 높은 문서 출현 빈도의 토큰 값을 기준으로 동일한 훈련 조건에서 고정 입력 길이만 다르게 하여 실험을 진행 하였다. 그 결과 상위 10개의 평균 토큰

수가 가장 높은 문서 분류 정확도를 보였다. 그러므로 본 논문은 신문 기사 데이터에 한하여 상위 10개의 평균 값 인 110을 고정 입력 길이로 한다.

상위 10개의 평균 토큰 수보다 적은 토큰 수를 가진 문서에 대해서는 적은 토큰 수만큼 0으로 채우는 zero-padding 방식을 활용하였고, 그 이상의 토큰 수를 가진 문서에 대해서는 상위 10개의 평균값으로 나누어 나눈 몫만큼 문서를 쪼개서 사용하였다. 상위 10개의 평균 토큰 수로 나누어지지 않아 나머지 토큰을 포함하고 있는 문서에 대해서는 다시 상위 10개의 평균 토큰 수만큼 0의 값으로 채웠다.

4.3 기존 연구와의 비교 실험 및 성능 분석

문서를 원하는 범주에 정확히 분류했는지 평가하기 위하여 분류의 정확도를 (수식 5)기준으로 측정했다.

$$Accuracy = \frac{Correctly\ classified\ documents}{Total\ number\ of\ documents} \quad (5)$$

precision, recall, F-measure의 점수를 기준으로 기반 모델과 제안 모델의 성능을 최종 비교 분석한다.

$$\mathrm{Precision} = \frac{Correctly\ classified\ documents}{Total\ documents\ predicted\ by\ model} \quad (6)$$

$$Recall = \frac{Correctly\ classified\ documents}{Total\ documents\ to\ be\ classified} \quad (7)$$

$$F-measure = \frac{2 \times \mathrm{Precision} \times Recall}{\mathrm{Precision} + Recall} \quad (8)$$

본 논문에서 제안하고자 하는 방식의 성능을 입증하기 위하여 문서 분류와 관련된 기존의 연구 방식과 비교 실험을 하였다. 표 2는 기존의 연구와 제안 방법에 대한 문서 분류 정확도를 비교한 표이며,

그림 6는 F-measure 점수 비교 차트이다. 이전 연구들의 성능보다 제안 모델이 평균적으로 12% 우수한 성능을 보인다. 문장 단위로 이루어진 데이터에서 우수한 성능을 보였던 CNN기반 연구[2]보다 영문장은 대략 16.%, 국어 문장은 3% 정확도가 증가했다.

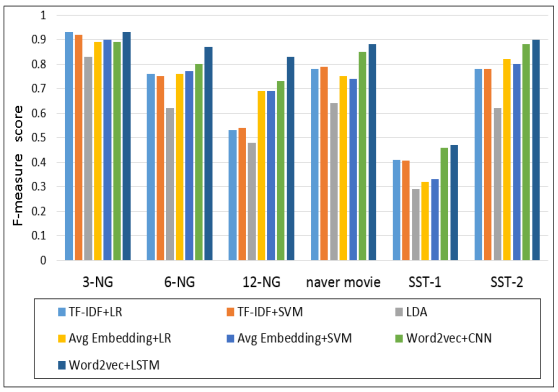

그림 5. 모델에 따른 f-measure 비교 차트
Fig. 5. A f-measure score comparison chart for each model

5. 결론 및 향후 연구

최근 자연어 처리 분야에서 주목 받고 있는 심층 신경망 알고리즘을 활용하여 효과적으로 문서 분류하고자 word2vec을 활용한 LSTM 모델을 새롭게 제안하였다. 기존의 자연어 처리 분야에서 우수한 문서 분류 성능을 보였던 학습 모델보다 성능이 더 향상되었으며, 가장 최근에 주목 받은 CNN 기반의 분류 모델[2]보다도 개선된 성능을 보여주고 있다. 또한, 새로운 방식으로 LSTM 고정 입력 길이를 결정하여 긴 구문의 문장 훈련 시 발생 할 수 있는 자원낭비 및 훈련시간 증가를 최소화 하여 긴 문장을 포함한 문서에서도 빠르고 정확한 훈련이 가능했다.

향후 과제로는 LSTM 층 및 fully connected 층을 추가하여 LSTM의 모델을 깊게 하고 모델의 매개변수를 튜닝하는 방법 등을 통한 성능 향상관련 연구를 지속할 계획이다.

References

[1] KIM, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882*, 2014.

표 2. 모델에 따른 테스트 데이터 정확도
Table 2. Accuracy of test data by model

Reference	Classification Techique	3NG	6NG	12NG	naver movie	SST-1	SST-2
A Genkin et al. [17]	TF-IDF + LR	0.93	0.76	0.53	0.78	0.40	0.78
H Drucker et al. [18]	TF-IDF + SVM	0.92	0.75	0.54	0.79	0.40	0.79
DM Blei al. [19]	LDA	0.83	0.62	0.48	0.64	0.29	0.62
Lai, Siwei, et al.[20]	Avg Embedding + LR	0.89	0.76	0.69	0.75	0.32	0.82
Lai, Siwei, et al.[20]	Avg Embedding + SVM	0.90	0.77	0.69	0.74	0.33	0.80
YoonKim et al. [1]	Word2vec + CNN	0.89	0.80	0.73	0.85	0.45	0.88
this work	Word2vec + LSTM	0.93	0.87	0.83	0.88	0.47	0.90

[2] Wang, Peng, et al. "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification." *Neurocomputing* vol. 174, no. 1, pp.806-814, 2016

[3] Dowoo Kim, Myoung-Wan Koo. "Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec", *Journal of KIISE*, vol.44. no. 7, pp.742-747, 2017

[4] In-Su Kang. A Comparative Study on Using SentiWordNet for English Twitter Sentiment Analysis. *Journal of Korean Institute of Intelligent Systems*, vol. 23, no. 4 , pp. 317-324, 2013.

[5] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, 2013.

[6] Mei-Ying Ren, Sinjae Kang. "Comparison Between Optimal Features of Korean and Chinese for Text Classification." *Joural of Korean Institute of Intelligent Systems*, vol. 25, no. 4, pp. 386-391, 2015.

[7] Dong-Wook Lee, Seo-Hyeon Baek, Min-Ji Park, Jin-Hee Park, Hye-Wuk Jung, Jee-Hyong Lee. "Document Summarization Using Mutual Recommendation with LSA and Sense Analysis." *Journal of Korean Institute of Intelligent Systems*, vol. 22, no. 5 , pp. 656-662, 2012

[8] Sunghae Jun. "A Big Data Preprocessing using Statistical Text Mining." *Journal of Korean Institute of Intelligent Systems*, vol. 25, no. 5, pp. 470-476, 2015

[9] Recurrent Neural Network(RNN) Tutorial-Part1, "Team AI Korea", Available: http://aikorea.org/blog/rnn-tutorial-1/, 2015, [Accessed: July 26 2017]

[10] Hochreiter,S. & Schmidhuber, J. "Long short-term memory" *Neural computation*, vol. 9, no. 8 , pp. 1735-1780, 1997

[11] Su Jeong Choi, Seong-Bae Park. "Categorization of POIs Using Word and Context information." *Journal of Korean Institute of Intelligent Systems*, vol, 24, no. 5, pp. 470-476, 2014

[12] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.

[13] mmihaltz, "word2vec-GoogleNews-vectors", Available:https://github.com/mmihaltz/word2vec-GoogleNews-vectors, 2016, [Accessed: July 2 2017]

[14] "The 20 Newsgroups data set", Available: http://qwone.com/~jason/20Newsgroups/ 2008, [Accessed: March 9, 2017]

[15] "Naver sentiment movie corpus v1.0", Available: https://github.com/e9t/nsmc 2015, [Accessed: July 9, 2017]

[16] "Stanford Sentiment Treebank", Available: https://nlp.stanford.edu/sentiment/ 2011, [Accessed: July 20, 2017]

[17] Genkin, Alexander, David D. Lewis, and David Madigan. "Large-scale Bayesian logistic regression for text categorization." *Technometrics* vol. 49, no. 3, pp. 291-304, 2007

[18] Drucker, Harris, Donghui Wu, and Vladimir N. Vapnik. "Support vector machines for spam categorization." *IEEE Transactions on Neural networks*, vol. 10, no.5, pp. 1048-1054, 1999

[19] BLEI, David M.; NG, Andrew Y.; JORDAN, Michael I. "Latent dirichlet allocation", *Advances in neural information processing systems*, 2002

[20]Lai, Siwei, et al. "Recurrent Convolutional Neural Networks for Text Classification." *AAAI*, vol. 333, no. 1, 2015.

저 자 소 개

김정미(Jung-Mi Kim)

2012년 : 동덕여자대학교 컴퓨터학 학사
2015년~현재 : 인하대학교 일반대학원 컴퓨터공학과 석사과정

관심분야 : Information retrieval, Text Mining, Deep Learning
Phone : +82-32-860-7453
E-mail : marine_k@naver.com

이주홍(Ju-Hong Lee)

1983년 : 서울대학교 전자계산기공학 학사
1985년 : 서울대학교 컴퓨터공학 공학석사
2001년 : 한국과학기술원 컴퓨터공학 공학박사
2001년~현재 : 인하대학교 컴퓨터공학부 교수

관심분야 : Artificial Neural Network & Deep Learning, Machine Learning
Phone : +82-32-860-7453
E-mail : juhong@inha.ac.kr