디지털산업정보학회 논문지 제15권 제1호-2019년 3월
https://doi.org/10.17662/ksdim.2019.15.1.087

# 워드 임베딩과 CNN을 사용하여 영화 리뷰에 대한 감성 분석*

주 명 길**·윤 성 욱***

# *Sentiment Analysis on Movie Reviews Using Word Embedding and CNN*

Ju Myeonggil·Youn Seongwook

〈Abstract〉

Reaction of people is importantly considered about specific case as a social network service grows. In the previous research on analysis of social network service, they predicted tendency of interesting topic by giving scores to sentences written by user. Based on previous study we proceeded research of sentiment analysis for social network service's sentences, which predict the result as positive or negative for movie reviews. In this study, we used movie review to get high accuracy. We classify the movie review into positive or negative based on the score for learning. Also, we performed embedding and morpheme analysis on movie review. We could predict learning result as positive or negative with a number 0 and 1 by applying the model based on learning result to social network service. Experimental result show accuracy of about 80% in predicting sentence as positive or negative.

Key Words : Social Network Service, Sensitivity Prediction, Morpheme Analysis, Embedding, Learning

## Ⅰ. 서론

사회적 관심이 증가하면서 사람들의 주요 정보공유 수단인 SNS(Social Network Service)의 활용도가 점점 증가하고 있는 추세이다. 동영상이나 사진, 유용한 정보를 등록하고 자신이 관심 있는 항목을 집중적으로 볼 수 있듯이 SNS의 기능도 많은 변화가 일어났다[1]. twitter나 facebook의 글을 보면, 어떤 제품이나 영화에 대한 평가를 했던 내용이 있다. 수많은 사용자들이 자신들의 감정을 표현하며 다른 사람들과 공유 및 소통을 가능하게 한다. 최근 빅데이터(Big-Data)의 등장으로 SNS에 대한 사용자의 글들이 중요한 분석 자료로 사용된다. 사용자들이 등록한 내용에는 다양한 감정이 존재한다. 특정 대상에 대한 호감이 있는 표현이나 거부감이 있는 표현이 감정에 대한 표본이다. 특정 대상에 대한 표현들은 시간이 지나면서 사용자에 의해 등록된다. 사용자들에 의해 등록된 내용이 점차 방대한 양의 빅데이터를 이루면서 긍정과 부정으로 대표적으로 나눌 수 가 있다.

사용자가 등록한 문장을 확인해보면 "좋다"와 "싫

* 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2017R1A2B1009139)
** 한국교통대학교 소프트웨어학과 학생
*** 한국교통대학교 소프트웨어학과 교수(교신저자)

다"를 구분할 수 있는 문장이 아니라"괜찮지만", "별로 이었지만"과 같이 흐름을 바꾸는 표현이 존재한다. 그렇기 때문에 단어가 가지는 의미가 중요한 것이 아니라 단어와 단어의 관계를 살필 필요가 있다. 한 문장에 긍정적인 단어와 부정적인 단어의 두 집합을 이용해 각각의 관계성을 나타내어 비교해보면 긍정과 부정을 판별할 수가 있다. 물론 긍정과 부정의 집합체를 구성하는 것은 큰 어려움이 있다. 구성하기 위해서는 많은 데이터를 수집해서 신뢰도를 쌓아야 한다. 한 개의 문장으로는 신뢰가 없어 추측에 그치지만 많은 양의 문장을 학습시키면 추측이 그렇다고 믿어지는 사실이 된다[2].

학습을 위해 첫째 작성된 글이 긍정적인 의미로 사용되었는지 부정적인 의미로 사용되었는지 판단할 수 있도록 많은 양의 학습 데이터를 수집하여야 한다. 본 논문에서는 누구나 쉽게 접할 수 있고 점수를 사용하여 어느 정도 긍정적인 성향을 가지는지 또는 부정적인 성향을 가지는지 확인할 수 있는 영화 리뷰를 대상으로 하였다. 영화리뷰는 영화에 대해 점수를 부여하고 긍정적인 말이나 부정적인 말을 사용하여 평가를 한다. 이를 이용하여 점수가 낮은 리뷰는 부정적인 말들이 사용되고 점수가 높은 리뷰는 긍정적인 말들이 대부분의 리뷰로 남아있는 것을 알 수 있다. 둘째, 문장에 형태소 분석기를 사용하여 단어로 나누어 학습에 필요한 단어와 불필요한 단어를 구성하였다. 문장에서 조사나 주어, 특수문자와 같은 언어는 긍정, 부정을 구분하기에 영향을 미치지 않는다고 판단하였다. 형태소 분석기는 자연어 처리에 있어 뛰어난 성능을 발휘하는 것으로 알려져 있는 python 제공 모듈을 사용하였다. 셋째 Word2Vec를 통해 단어를 수치화하여 각 단어에 가중치를 할당할 수 있도록 단어벡터를 구성한다. 수치화 과정 후 생성된 모델을 이용하면 문장을 구성하는 각 단어는 점수를 갖게 된다. 각 단어의 점수들은 기계학습을 진행하는데 사용된다.

본 논문은 다음과 같이 구성하였다. 제 2장에서는 소셜 네트워크와 문장 분석에 관한 관련 연구 및 연구에서 사용된 Word2Vec를 소개한다. 제 3장에서는 TensorFlow로 데이터를 학습시키는 과정과 학습에 사용된 방법론을 소개한다. 제 4장에서는 학습 모델을 사용하여 도출된 결과와 성능에 대한 결과를 소개한다. 제 5장에서는 본 연구의 결론과 방향성을 제시하였다.

## Ⅱ. 관련연구

소셜 네트워크를 분석한 연구 사례 중 [3]의 연구에서는 SNS 사용자들이 대학에 대해 작성한 글을 분석하여 대학에 대한 인식과 선호도를 분석하였다. 또한, 하둡 기반의 언어와 툴을 사용하여 수집된 데이터에 대한 다차원 분석을 수행하였으며 이를 통해 대학과 관련된 언어에 대한 연관성을 예측하였다.

사용자의 성향을 이용한 TV 광고 분석 연구[4]에서는 [3]의 연구와 마찬가지로 하둡과 오피니언 마이닝 기법을 사용하여 신뢰도에 대한 분석 정확도록 높였다.

한편, 워드임베딩과 그래프 기반 준지도 학습을 통한 한국어 어휘 감성 점수를 산출하는 연구[5-6]에서는 영화 댓글을 학습 데이터로 사용하였다. 사이트의 영화 댓글을 수집하여 Word2Vec에 의해 단어를 수치화 시켰다. 단어 임베딩을 통해 각 단어들은 임베딩 공간에서 거리기반의 네트워크를 이루었다. 여러 가지 방식 중에 CBOW(Continuous Bag of Word)와 skip-gram 두 개의 방법을 사용하여 진행하였다. 가중치를 적용하기 전에 명확한 단어인 pre-label을 선정하여 그래프 기반 준지도 학습을 통해 감성사전을 구축하였는데, 그래프 기반 준지도 학습은 RBF(Radial Basis Function) Kernel을 사용하여 단어 네트워크에 연결된 두 단어 사이의 거리에 따른 가중치를 부여하는 방식을 사용하였다.

# Ⅲ. 관련모델

## 3.1 형태소 분석기

Python 언어에서는 자연어를 처리하기 위해 다양한 형태소 분석기를 제공한다. 주로 사용되는 모듈은 NLTK(The Natural Language Toolkit), kkma, twitter 분석기 이다. 본 논문에서는 각 분석기들의 기능과 처리 시간의 분석을 통해 본 연구의 목적에 가장 적절한 분석기를 선택하여 사용하였다.

NLTK는 주로 영어문장을 분석할 때 사용된다. 영어문장을 분석하게 되면 띄어쓰기를 기준으로 각 단어의 품사가 잘 나뉘어져 있다. NLTK모듈을 통해서 제공하는 말뭉치(Corpus)를 사용할 수 있다. 한글 문장을 분석할 경우에는 띄어쓰기를 기준으로 분석을 하는 경향이 보여 띄어쓰기로 구분된 단어는 조사와 함께 붙여져 있다. 이런 이유로 형용사와 동사, 명사로 구분하기가 어렵다[7].

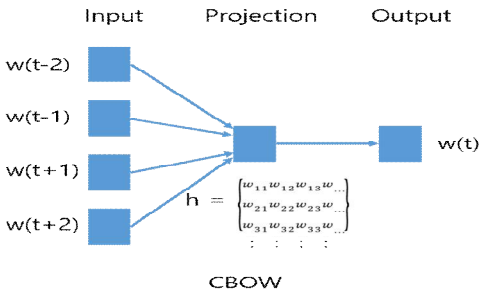
한글문장을 분석하는데 사용되는 모듈은 kkma와 twitter모듈이 존재한다. 한글 문장을 분석하는 모듈은 C/C++, Java에서 개발되어 왔다. 하지만, Python에서는 모듈을 사용하는 것이 편리하고 무엇보다 언어의 특성상 이해하기가 쉽다. 꼬꼬마 분석기는 한글 문장 분석이 정말 뛰어나지만, 추후에 학습을 수행하고 데이터를 정제하는데 많은 시간이 걸리게 된다. 조사, 명사, 형용사 등뿐만 아니라 어말 어미의 형태소까지 분석하기 때문에 이 점까지 고려하면서 연구를 수행하는 것은 오랜 시간과 좋은 결과를 기대하기가 어렵다[8].

twitter분석기는 단어단위로 필요한 부분과 필요없는 부분을 잘 분석하는 것을 볼 수 있었다. 분석할 때 사용하지 않는 주어나 조사, 특수문자까지 확인할 수가 있어서 불용어 사전을 구축하기에 편리했다. 영어문장의 분석결과는 띄어쓰기가 되어 있는 대로 모든 분석기가 품사의 정보를 제외하고 같게 분석하였다. 한글의 경우에는 주로 조사를 구분할 수 있는 부분에서 차이가 났다.

NLTK는 단어에 대한 정보보다는 띄어쓰기의 중점으로 분석되었고, kkma와 twitter는 단어의 의미에 따라 분석이 잘 되었지만, kkma의 경우에는 필요 없는 부분까지 분석을 수행하여 전처리를 하는 것이 더 까다로웠다. 이뿐만 아니라 문장 수에 따라 분석을 완료하는 시간도 차이가 났다. 문장을 이루는 단어의 수가 많으면 많을수록 kkma가 twitter분석기 보다 훨씬 오랜 시간이 소모되었고, 적은 단어로 이루어진 문장은 큰 차이가 없었다. 따라서 본 논문에서는 분석기능이나 처리시간으로 볼 때 twitter분석기가 다른 분석기에 비해 효율이 좋기 때문에 twitter분석기를 사용하였다.

## 3.2 Word2Vec

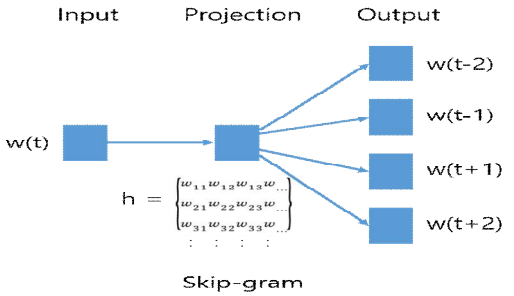
Word2Vec는 문장에서 사용된 단어들이 서로 관련이 되어있는 관계 정도를 정의하며 단어의 유사성과 의미에 따라 값이 정의된다. Word2Vec의 대표적인 학습 모델은 CBOW와 skip-gram이 있다. CBOW모델은 문장을 기반으로 단어를 예측을 하는데 사용되어지고 skip-gram은 한 단어를 기준으로 문장을 구성할 수 있는 주변 단어를 예측하는 모델이다.




<그림 1> CBOW Diagram

CBOW모델의 작동원리는 <그림 1>과 같다. 먼저, 입력층(Input Layer)에 연속적인 문장이 입력된다. 입력된 문장은 투사층(Projection Layer)에서 가중치 매트릭스를 이용하였으며 연산을 수행한 뒤에 수치화되어 구성된다. 위에서 예로든 "화려한 배경과 뛰어난 배우들의 연기가 정말 좋았습니다."라는 문장에 모델링을 적용하게 되면, "화려한 배경과"에서 "뛰어난", "배우들"과 같이 문장을 구성하는 단어를 예측할 수가 있다[9].

skip-gram 모델은 CBOW와는 달리 문장이 주어졌을 때 문장의 구성을 이루는 단어들 중에 임의의 한 단어를 기준으로 단어의 앞, 뒤에 올 단어를 예측할 수 있어야 한다.



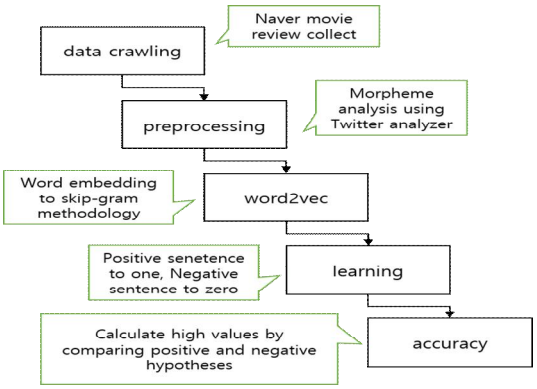

<그림 2> skip-gram Diagram

skip-gram은 <그림 2>와 같이 입력 층에서는 단어로 구분된 문장이 입력된다. 투사 층에서는 가중치 매트릭스를 통해 단어가 숫자로 이루어진 n-차원의 벡터가 구성된다. 적용 예를 들면, "화려한 배경과 뛰어난 배우들의 연기가 정말 좋았습니다."라는 문장에서 "뛰어난"이라는 단어에 다음에 올 수 있는 "배우들의"라는 단어를 예측하게 된다. 각 단어는 위치를 가지고 있기 때문에 그래프를 통해 주변에 어떤 단어들이 위치하고 있는지 확인할 수가 있다[10].

## Ⅳ. 데이터 수집과 워드 임베딩 및 데이터 학습과정

### 4.1 연구절차

우선, 데이터를 수집할 때, Word2Vec를 수행할 문장들과 TensorFlow를 이용하여 학습할 데이터를 따로 설정하였다. Word2Vec의 경우에는 단지 단어들을 수치화 시키는 역할이기 때문에 많은 어휘를 포함하여야한다. 그렇기 때문에 가능한 많은 문장들을 포함하여야 하며, 많은 문장을 학습하게 되면 어휘력이 풍부하기 때문이다. 연구를 진행하는데 있어서 <그림 3>은 본 연구의 전체적인 처리과정을 나타낸 것이다. 먼저 감정분석에 필요한 데이터를 수집해야 한다. 수집된 데이터 중 모든 단어가 사용되지 않기 때문에 전처리 과정을 통해 필요한 데이터를 분류한다. 마지막으로 분류된 데이터를 이용하여 학습을 수행하고 학습 모델을 생성하여 정확도를 측정한다. 데이터 수집은 접근이 쉽고 전처리를 수행하는데 편리한 네이버 영화로 선정하였다. Word2Vec로 학습할 데이터는 사람들이 블로그에 주로 자신의 느낌으로 자세하게 설명한 부분이다. 그렇기 때문에 많은 양의 문장들을 가져올 수 있고 양도 1,000,000개가 넘는다.




<그림 3> Flowchart

그리고 TensorFlow를 이용하여 긍정과 부정을 예측할 데이터는 영화 리뷰중 짧게 자신의 느낌을 작성한 댓글을 선정하였는데, 수집할 수 있는 페이지가 1,000페이지밖에 되지 않아서 4일에 한 번씩 수집하여 약 한 달 동안 총 7,000페이지를 수집하였다. 댓글과 함께 평점이 작성되어 있어서 긍정과 부정을 구분하기에 좋은 조건을 갖추었다. 수집된 데이터의 전처리는 python에서 제공하는 형태소 분석기를 사용하였으며, Word2Vec의 방법론 중에 skip-gram을 사용할 수 있도록 문장을 단어 단위로 나누었다. 마지막으로 TensorFlow를 이용한 학습은 Word2Vec로 수치화된 단어 모듈을 이용하여 학습할 단어들의 관계를 모두 숫자로 바꾸어 저장시킨 뒤에 각 단어들의 관계를 부정이나 긍정의 결과에 부합시켰다. layer에 따라 수행되며, 각 수행과정마다 높은 cost값을 산출해 결과 값과 비교하였다. 각 비교된 결과를 누적하여 총 정확도를 측정하였다.

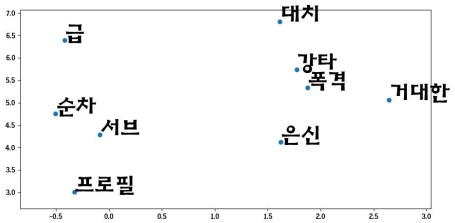
## 4.2 데이터 수집과 전처리

Word2Vec 모델을 적용하기 전에 학습에 사용할 데이터는 네이버 영화 리뷰의 댓글을 대상으로 하였다. 데이터 수집은 Python을 사용하여 수집하고자 하는 페이지의 url을 여는 것부터 시작한다. url을 열게 되면 html의 소스코드가 추출된다. 이 소스코드에서 사용자가 작성한 댓글과 평점을 가지고 있는 태그를 find함수 또는 select함수로 가져와 데이터를 수집하여 텍스트 파일로 저장하였다.

약 1,100,000개의 학습 리뷰 중에 1-4사이의 평점은 부정적인 언어가 많이 사용되었을 것이라 판단하였고 7-10사이의 평점은 긍정적인 언어가 많이 사용되었을 것이라 판단하여 2가지로 나누었다. 1,100,000개의 학습 리뷰 모두 사용할 수 있는 데이터가 아니기 때문에 데이터의 선정과정에서 평점이 5점, 6점인 데이터와 인코딩에 문제가 있는 데이터를 제외하고 negative 데이터의 수는 485,522개이고 긍정 데이터의 수는 218,584개이다. 부정 리뷰가 상대적으로 많았다.

Word2Vec이외에 TensorFlow로 학습을 진행할 학습 데이터도 수집하였다. Word2Vec의 학습 데이터는 많은 단어를 포함해야하는 반면에 TensorFlow에 적용할 학습 데이터는 자신의 감정이 잘 나타나있는 데이터를 사용해야하기 때문에 영화 리뷰 중 영화에 대한 평가만 작성한 리뷰를 수집하였다. 데이터를 수집하는 과정에 있어서 불필요한 문자들이 존재하기 때문에 임의로 불필요한 문자들을 분석대상에서 제외시켰다[11].

각 학습 데이터들은 순서대로 twitter 형태소 분석기를 통해 단어의 형태소에 맞게 분석을 실행하였다. 문장 단위로 리스트에 저장되며 그 문장은 단어로 다시 구분된다. 리뷰의 수만큼 공간을 차지한 리스트는 함수의 설정 값에 따라 학습이 진행된다. 모듈 안에 변경을 준 사항은 벡터의 사이즈와 윈도우 크기, 최소 빈도수, CBOW와 skip-gram중 학습할 방법, 코어의 개수, 학습 epoch이다. 벡터의 사이즈는 문자가 이루는 숫자의 사이즈로 100으로 두었다. 윈도우의 크기는 2로 하였는데, 2는 앞뒤 연속하는 단어에 중점을 두었다. 학습 방법은 skip-gram으로 하였고 쿼드코어를 사용하여 100번의 학습을 하였다.




<그림 4> Word Vector Sample

<그림 4>는 수치화된 단어의 일부분을 나타낸 것인데, 100차원으로 된 것을 출력을 위해 2차원 x, y값

으로 바꾸어 나타내었다. x축과 y축은 각 단어의 위치를 나타내며 단어의 거리가 가까울수록 유사한 의미를 가진다. 29,613개의 단어들이 이렇게 구성되어있어서 테스트문장에 적용하게 되면 값을 가진 문장으로 만들 수가 있다. 그러나 중요한 점은 문장이 아니라 어떤 긍정이나 부정이라는 문장을 구성하는 단어 사이의 관계가 중요하기 때문에 긍정문장일 경우에는 1이라는 값을 배정하고 그 문장을 구성하는 단어들의 관계 역시 1이라고 정하였다. 부정의 경우에는 0이라는 숫자를 같은 형식으로 적용하였다. 구성 결과로는 2:1의 비율로 긍정적이면 "단어:1", 부정적이면 "단어:0"을 이루는 배열이 생성된다. 이제 이 배열을 TensorFlow로 학습시켜 문장을 판별하는 과정을 수행하였다.

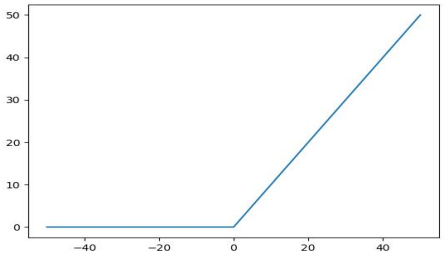
## 4.3 CNN을 이용한 데이터 학습

본 논문에서는 심도 있는 학습을 위해 Deep Learning의 한 종류인 CNN(Convolution Neural Network)의 방식을 적용한 딥러닝을 수행하였다[12].

CNN의 수행절차는 정의된 layer에서 합성 곱 연산을 수행한다. 학습에 준비된 데이터는 2가지의 입력과 출력으로 구성되어있고, 입력으로는 수치화된 두 단어들의 집합과 출력으로는 긍정과 부정을 나타내는 0과 1의 집합으로 이루어져있다. 본 연구에서는 5개의 layer를 사용하여 진행하였다. layer의 기능은 식(1)과 같이 수치화된 단어와 W(가중치)를 곱하여 바이어스를 더한 형태를 취하게 된다. 위의 Word2Vec에 나타난 그림과 마찬가지로 연산을 수행을 거듭하면서 출력 값에 수렴할 수 있도록 설정한다.

$$L(i) = X * W(i) + b \tag{1}$$

각 layer는 식(1)의 형식으로 연산을 수행한다. 처음 이후에 수행되는 layer의 연산은 입력 값 X대신의 이전 layer의 결과가 사용된다. 연산결과는 굉장히 광범위한 분포를 이루게 된다. 규칙성이 없는 분포는 결과물로 사용하기 어렵기 때문에 값을 일정범위내로 표현할 수 있도록 Sigmoid나 ReLU(Rectified Linear Unit)과 같은 작업을 통해 결과 값을 0-1사이로 표현하거나, 음수는 0으로 양수는 그대로 표현하는 방법을 사용하였다[12].




<그림 5> ReLU Distribution Chart

ReLU의 분포도는 <그림 5>와 같이 나타나게 된다. 이 방법은 보통 layer의 개수가 많을 경우에 사용한다. x축은 각 단어를 나타내고 y축은 단어에 대응되는 값으로 오름차순 정렬하였다. 그 이유는 layer의 수가 많아지면 연산의 양이 그만큼 늘어나게 된다. Sigmoid로 계산할 경우에는 0-1의 범위에 값이 포함된다. layer의 수와 데이터의 수가 적을 때는 Sigmoid함수를 사용하여 0-1사이에 존재하는 다양한 수를 확인할 수 있지만, 굉장히 많은 데이터와 layer의 수가 많아지면 연산 결과의 폭이 커진다. layer마다 Sigmoid를 계속해서 사용하게 되면 0과 1에 수렴하는 수가 많아져 정확한 결과를 얻을 수 없다. 이런 문제의 해결책이 바로 ReLU를 이용하여 연산의 결과 값들이 양수이면 그대로 표현하고 0보다 작은 수는 0으로 수렴할 수 있도록 max(0, X)의 형태의 연산을 수행하는 것이다. 즉, 0과 X값 중 최댓값을 정한다[12].

모든 layer를 거쳐 나온 결과 값은 출력형식에 맞게 cost값을 산출해 주는 연산을 수행한다. 출력 결과에 사용된 라벨은 Y값으로 긍정과 부정을 나타낸다. 긍정을 1로 정하고 부정을 0으로 정하였기 때문에 연산 결과가 두 개의 cost값으로 축소된 뒤에 더 큰 값의 index를 출력하게 된다. 위에서 설명했듯이 0의 자리는 부정을 나타내고 1의 자리는 긍정을 나타낸다. 단어와 단어사이의 관계가 부정이면 0이 출력되고 긍정이면 1을 출력하게 된다.

$$cost(W) = -\frac{1}{m}\sum y\log(H(x)) \quad (2)$$

softmax_cross_entropy_with_logits함수를 이용하여 모든layer의 연산을 통해 연산된 값을 log함수를 취해 각 입력의 출력 값인 Y값에 대응 시킨다. 식(2)에서 H(x)의 값은 모든 마지막 layer를 제외한 모든 layer를 수행한 마지막 layer에서 ReLU를 적용하지 않은 값이다. 각 수행과정을 모두 더해 평균값으로 나누면 cost값이 계산된다. 각 수행과정마다 산출된 cost값을 값들의 분포는 감소하는 형태를 취하게 된다. 이 과정에서 손실이 발생할 수 있기 때문에 cost값들을 각 수행과정마다 일정 학습 비율을 이용하여 손실을 최소화하기 위해 최적화 과정을 통해 판별을 할 수 있도록 긍정과 부정을 함수로 나타낸다.

입력 데이터는 200차원의 두 단어가 임베딩된 값들의 집합이 입력되고 출력 데이터는 단어사이의 관계가 긍정인지 부정인지를 나타내는 0과 1이 출력되도록 대략적인 학습 과정이 나타나 있다. skip-gram으로 설계하여 한 문장에서 나타나는 모든 단어들의 이웃관계로 배열을 이루게 된다. 연구의 목적이 문장 전체를 보고 예측을 하는 것이 아니라 문장을 구성하는 단어의 관계를 통해 예측을 하는 것이기 때문에 한 단어로 쪼개어진 문장을 연속하는 두 단어로 관계를 다시 형성하는 작업을 통해 학습이 진행되었다[5].

# Ⅴ. 실험에 대한 결과

## 5.1 Word2Vec의 결과

댓글을 사용한다는 점이 긍정적인 말과 부정적인 말을 구별할 수 있었다. skip-gram의 방법으로 단어를 분류하여 단어 임베딩을 수행하였다. 학습으로 수행된 결과를 확인해 보았을 때, 단어 사이의 관계 및 동의어를 다음과 같이 나타내었다.

<표 1> Similar Word Extraction

| Word | Similar Word and Accuracy |
|---|---|
| 행복 | ('행복함', 0.7173), ('기쁨', 0.7083)<br>('즐거움', 0.6715), ('쾌락', 0.6629) |
| 슬픔 | ('고통', 0.7021), ('괴로움', 0.6637)<br>('비극', 0.6608), ('죽음', 0.6382) |

<표 1>는 Word2Vec에서 제공하는 most_similar함수를 이용한 것인데, 행복과 슬픔을 기준으로 동의어를 출력한 것이다. most_similar함수는 단어 사이의 코사인 유사도를 이용하여 유사한 단어를 추출하는 함수이다. <표 1>에 제시된 단어 이외에 유사성이 낮은 단어들도 포함되어 있는데, 그 단어들 중에 유사성이 높은 단어들을 내림차순으로 4개 까지 출력한 것이다[13]. 행복을 입력했을 때 동의어로 행복함이 71%, 기쁨이 70% 그리고 슬픔을 입력했을 때 고통이 70%, 괴로움이 66%로 주변에 연관된 단어가 있다는 것을 확인할 수 있다[5].

## 5.2 학습 결과

Word2Vec로 학습된 단어들의 위치 값은 학습과정에 따라 연산된 다차원 값에 그친다. CNN을 통해 긍정과 부정으로 나뉜 단어들을 이용하여 일정한 규칙성을 갖게 하여 판별하는 것이 궁극적인 실험 목표이다. 학습에 사용된 학습 데이터는 총 651,562개의 단

어로 구성되며 18,940개의 문장에서 추출하였고 테스트를 위해 사용한 데이터는 총 103,876개의 단어로 구성되며 총 3,000개의 영화 리뷰에서 추출된 단어이다. 긍정문장과 부정문장을 동일한 비율로 적용하였다[14].

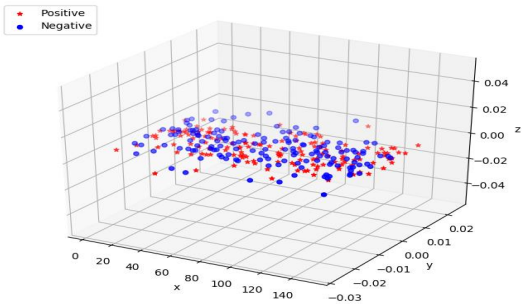
<그림 6>은 기계학습이 수행되기 전에 Word2Vec를 활용하여 수치화된 단어들의 합을 평균으로 표현한 것이다. 약 200개의 문장만 나타내었다. x축은 문장의 총 개수이고 y축은 x값에 대응하는 문장의 평균값이다. 그래프를 통해서 볼 수 있듯이 위치 값으로는 긍정과 부정문장의 분포가 균등하게 나타나기 때문에 문장을 구분하기가 어렵다. 그래서 기계학습을 이용하여 layer마다 가중치를 곱하여 차원을 변경시키고 바이어스를 더하는 것과 같은 연산을 거쳐 긍정과 부정을 판단할 수 있게 되었다.

layer의 연산 과정을 거치면서 <그림 7>처럼 가운데 선을 기준으로 긍정과 부정의 문장들이 일정 범위 영역의 값을 가지게 된다[15]. 본 연구에서는 긍정적 견해의 문장들을 1로 매칭 시켰고 부정적 견해의 문장들을 0으로 매칭 시켰다. 기계학습을 이용하여 평점을 기준으로 7-10점을 긍정문으로 1-4점을 부정문으로 가정한 뒤에 학습을 진행하여 그림과 같이 100%로 분류를 할 수 없지만, 약 85%의 정확도를 가질 수 있도록 학습이 되었다.

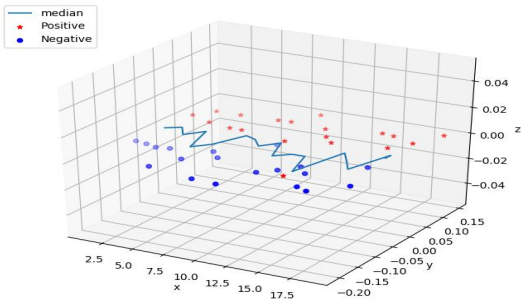
## 5.3 예측 결과

예측대상은 모델을 훈련시키고 긍정과 부정을 나눌 수 있는 기준이 뚜렷한 영화의 리뷰로 선정하였다. 그리고 예측과정은 twitter에서 수집한 데이터를 이용하여 나타내었다. 영화의 리뷰는 총 9,206개의 리뷰를 수집하였으며, 7~10점의 점수를 나타낸 리뷰를 긍정으로 선정하였고 1~4점의 점수를 나타낸 리뷰를 부정으로 선정하였다. 수집결과 긍정 리뷰의 개수는 7,640개이고 약 83%의 비중을 차지한다. 부정 리뷰의 개수는 1,562개이고 약 17%의 비중을 차지한다.

이 영화의 리뷰를 기준으로 모델을 적용한 결과 긍정 리뷰의 개수는 6,536개로 약 71%의 비중을 차지하고 부정 리뷰의 개수는 2,670개로 약 29%의 비중을 차지하여 약 85%의 정확도를 나타내었다. 기존연구의 방식은 그래프 기반 준지도 학습을 통해 감성 점수를 산출하는 방식을 적용하였다. 감성 사전을 구축하여 형태소 분석기로 나뉜 단어들은 임베딩을 통해 값을 가지게 된다. 각 단어들을 노드라 하며 네트워크를 생성하게 된다. 이 네트워크에 Label Propagation과 Label Spreading을 적용하여 긍정점수와 부정점수를 계산하였다. -1에서 +1사이의 점수를 갖게 되며 -1에 가까우면 부정으로 예측하고 +1에 가까우면 긍정으로 예측하였다[5]. 본 연구에서는 점수를 계산하는 방법 대신에 label을 이용하여 두 단어사




<그림 6> Sum of Word Vector Before Learning




<그림 7> Sample of Learning Result

이의 관계가 긍정이면 1로 예측되고 부정이면 0으로 예측하였다. 단어마다 점수를 계산하는 방법과 다르게 단어 사이의 관계를 학습시키는데 목표를 하였다.

<표 2>은 SNS에 임의의 검색 대상을 선정하여 검색한 뒤에 검색 결과를 수집하여 TensorFlow를 실행한 결과이다. <표 2>의 데이터는 twitter API를 사용하여 수집하였으며, 전처리 작업에서 사용하였던 방법과 마찬가지로 예측에 필요하지 않은 문장들은 제외시켰다. 데이터를 수집하는 과정은 검색 하고자 하는 키워드를 입력한 후에 검색 기간을 설정하여 허용되는 양의 데이터만 수집할 수 있다. 검색 대상은 '연출'이라는 키워드를 입력하였고 검색 기간은 기본 설정대로 두어 최근에 작성된 문장을 수집하였다. 분석 대상의 결과를 보면 긍정과 부정으로 나뉘어져 있는데, 일부 문장만 가져온 결과이다.

분류된 단어는 인접한 두 단어로 이루어져있으며 각 단어의 집합은 학습된 모델을 통해 결과를 나타내었다.

긍정으로 예측한 문장은 '멋진 모습이 연출 될 거 같아'이다. <표 2>에서 형태소에 따라 '[멋진, 모습]', '[모습, 이]'와 같이 문장을 구성하는 단어의 쌍으로 예측을 하였다. 부정문장도 마찬가지로 수행하였으며, 단어의 쌍이 긍정이면 1로 예측하였고 부정이면 0으로 예측하였다. 1과 0의 개수를 카운트하여 1의 개수가 0의 개수보다 많은 경우 긍정문장이 되고 0의 개수가 1의 개수보다 많은 경우 부정문장이 된다. 그 결과 '멋진 모습이 연출 될 거 같아'이라는 문장은 1의 개수가 0의 개수보다 더 많기 때문에 긍정문장으로 예측하였고 '이 연기는 납득이 안 된다'라는 문장은 0의 개수가 1의 개수보다 더 많기 때문에 부정문장으로 예측하였다.

## 5.4 학습 모델의 적용 결과

학습된 모델의 성능을 평가하기 위해 영화를 비교 대상으로 선정하였다. 다른 소셜 네트워크에 존재하는 데이터를 사용할 경우 해당 데이터가 긍정이나 부정이라고 판단할 수 있는 근거가 없다. 따라서 기존에 수집한 데이터와 같은 방식으로 2017년에 개봉한 5개의 영화를 선정하여 평점이 1점-4점을 부정적 데이터로 7점-10점을 긍정적 데이터로 선별하였다[16]. 선정 기준은 다른 국내 영화에 비해 댓글의 수가 많고 사용자들의 관심이 더 높은 영화를 선정하였다. 리뷰는 현재 시간을 기준으로 차례대로 수집하였다.

각 영화 별로 약 10,000개의 리뷰를 수집하여 평점에 따라 1, 0, 5라는 라벨을 붙였다. 평점이 7-10점이면 1이고 1점-4점은 0, 그리고 그 외의 평점을 가진 리뷰는 5로 하였다. 예측 방식은 <표 3>과 <표 4>에서 문장을 단어 관계로 나누어 1 또는 0으로 구분한 것과 같이 각 리뷰를 연속된 단어로 나누어 예측하였다.

<표 3>은 영화 리뷰 중 평점이 7-10점인 긍정 리뷰를 예측한 결과를 나타낸 것이다. 각 영화별로 7-10점 사이의 리뷰를 수집하여 수집된 리뷰의 개수와 모델을 적용하여 예측된 리뷰의 개수와 정확도를 측정하였다. 예측과정은 <표 2>와 같은 방식으로 예측하였으며 정확도는 7-10점의 리뷰 개수를 100%라 가정하고 모델을 적용하여 긍정으로 예측된 리뷰 개수만을

<표 2> Analysis Result of SNS Comment

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Positive Sentence | Classification | [멋진, 모습] | [모습, 이] | [이, 연출] | [연출, 될] | [될, 거] | [거, 같아] |
| | Prediction | 1 | 1 | 1 | 0 | 1 | 0 |
| Negative Sentence | Classification | [이, 연기] | [연기, 는] | [는, 납득] | [납득, 이] | [이, 안된] | [안된, 다] |
| | Prediction | 0 | 0 | 1 | 0 | 0 | 0 |

<표 3> Prediction Result of Positive Review

| | The King | A Taxi Driver | Confidential Assignment | Along With the Gods: The Two Worlds | The Battleship Island |
|---|---|---|---|---|---|
| number of grade 7-10 | 8,083 | 8,994 | 7,870 | 6,567 | 5,404 |
| number of prediction | 5,693 | 7,513 | 5,914 | 4,822 | 3,630 |
| accuracy (%) | 70.5 | 83.6 | 75.2 | 73.4 | 67.2 |

<표 4> Prediction Result of Negative Review

| | The King | A Taxi Driver | Confidential Assignment | Along With the Gods: The Two Worlds | The Battleship Island |
|---|---|---|---|---|---|
| number of grade 1-4 | 989 | 536 | 1,237 | 2,399 | 3,375 |
| number of prediction | 860 | 416 | 1,125 | 2,151 | 2,678 |
| accuracy (%) | 87 | 77.6 | 91 | 89.7 | 79.7 |

이용하여 정확도를 측정하였다.

<표 4>는 <표 3>과 동일한 영화로 평점이 1-4점인 부정 리뷰를 측정하였다. 1-4점의 모든 리뷰를 수집한 리뷰의 개수와 모델을 적용하여 예측된 리뷰의 개수와 정확도를 측정하였다. <표 3>의 긍정 리뷰에 비해 대체적으로 높은 정확도가 측정되었다.

## VI. 결론

본 연구에서는 기계학습을 이용하여 소셜 네트워크의 검색 대상에 대한 긍정과 부정을 식별하는 방법을 제안하였다. 본 연구에서는 제안 방법을 이용하여 10,000개의 영화 리뷰 데이터에 대한 긍정 및 부정 표현의 예측 정확도를 측정하였으며 평균 80%의 결과가 나왔다. 본 연구의 정확도가 연구[5]에 비해 비교적 낮지만, 예측과정에서 차별성이 존재한다. 본 연구에서는 문장을 형태소로 구분하여 연속되는 단어 쌍을 형성하였다. 본 연구의 모델은 구분된 단어 쌍을 각각 예측하기 때문에 결과의 과정을 자세히 알 수 있다. TensorFlow를 이용하여 학습을 통해 긍정과 부정을 예측하는 방법은 생각보다 쉽고 높은 기대치를 가져왔다. 그러나 학습 대상이 영화의 리뷰이다 보니 단어의 한계가 존재한다. 중복되는 단어나 영화제목이나 영화를 평가할 때 사용하는 단어와 같이 이러한 단어들만 학습하다보니 다른 이슈에 대해서 예측을 할 때 취약점이 보인다. 따라서 학습된 모델을 이용하여 SNS를 분석한 후 분석된 결과를 다시 학습을 통해 반복적으로 학습한다면 보완할 수 있다. 감성분석은 대중들의 반응을 중요시 한다. 따라서 본 연구를 이용하여 특정 주제에 대해 대중들의 답변을 긍정과 부정으로 예측하여 전체의 흐름을 알 수 있게 된다.

## 참고문헌

[1] 유혜림・송인국, “웹 서비스 형태 변화에 따른 소셜 네트워크 서비스의 진화,” 인터넷정보학회지, 제11권, 제3호, 2010, pp.52-62.

[2] 장환석・장은영・정광용, “Word2Vec를 이용한 감성어 분석 방법,” 한국정보과학회 한국소프트웨어종합학술대회 논문집, 2017, pp.661-663.

[3] 양민혁・정인선・김용채・조완섭, “SNS 데이터를 활용한 국내대학 인식 및 선호도 분석,” 한국빅데이터서비스학회, 제1권, 제1호, 2014, pp.1-13.

[4] 이아름・방지선・김윤희, “SNS Big-data를 활용한 TV 광고 효과 분석 시스템 설계,” 정보과학회 컴퓨팅의 실제 논문지, 제21권, 제9호, 2015,

pp.579-586.

[5] 서덕성・모경현・박재선・이기창・강필성, “워드임베딩과 그래프 기반 준지도학습을 통한 한국어 어휘 감성 점수 산출,” 대한산업공학회지, 제43권, 제5호, 2017, pp.330-340.

[6] 김윤석・서영훈, “기계 학습을 이용한 한글 텍스트 감정 분류,” 한국엔터테인먼트산업학회, 2013, pp.206-210.

[7] Bird, S. and Loper, E., “NLTK: The Natural Language Toolkit,” Cornell University, 2002.

[8] 박보국, “Python 환경에서 한글 형태소 분석기 패키지 KoNLPy 사용법,” 부산대학교, 2014.

[9] Mikolov, T. and Chen, K. and Corrado, G., “J. Dean, Efficient Estimation of Word Representations in Vector Space,” Cornell University, 2013.

[10] 이동훈・김관호, “Word2Vec 기반의 의미적 유사도를 고려한 웹사이트 키워드 선택 기법,” 대한산업공학회, 제2017권, 제211호, 2017, pp.923-938.

[11] 이경택・김창욱, “Word2vec과 shrinkage 회귀를 이용한 semi-supervised 감성 분석 기법,” 한국경영과학회, 2017, pp.3691-3707.

[12] 한정수・곽근창, “컨벌루션 신경회로망과 ReLU 함수 기반 ELM 분류기를 이용한 영상 분류,” 한국정보기술학회논문지, 제15권, 제2호, 2017, pp.15-23.

[13] 김지영・한다현・김종권, “빅데이터 검색 정확도에 미치는 다양한 측정 방법 기반 검색 기법의 효과,” 정보과학회논문지, 제44권, 제5호, 2017, pp.553-558.

[14] 고장혁・이동호, “정보 유출 탐지를 위한 머신 러닝 기반 내부자 행위 분석 연구,” 디지털산업정보학회, 제13권, 제2호, 2017, pp.1-11.

[15] 박흠・이창범, “기계학습 기반의 주행중 운전자 자세교정을 위한 지능형 시트,” 디지털산업정보학회, 제13권, 제4호, 2017, pp.81-90.

[16] 네이버 영화, https://movie.naver.com/movie/point/af/list.nhn

■ 저자소개 ■

주 명 길
(Ju Myeonggil)

2019년 한국교통대학교 소프트웨어전공학사 예정

관심분야 : 네트워크 개발, 빅데이터, 머신러닝
E-mail : echosoul1994@naver.com

윤 성 욱
(Youn Seongwook)

2015년 9월~현재 한국교통대학교 전자계산학과 교수
2012년 2월 LG전자 CTO연구원
2009년 2월 Ph.D. Computer Science, University of Southern California
2002년 2월 M.S. Electrical Engineering, University of Southern California.
1997년 2월 서강대학교 컴퓨터공학과(학사)

관심분야 : 데이터 분석 및 예측, 온톨로지, 센서 네트워크
E-mail : youn@ut.ac.kr

논문접수일 : 2019년 01월 14일
수 정 일 : 2019년 03월 05일
게재확정일 : 2019년 03월 08일