

Political Opinion Mining from Article Comments using Deep Learning

Dae-Kyung Sung*, Young-Seob Jeong**

Abstract

Policy polls, which investigate the degree of support that the policy has for policy implementation, play an important role in making decisions. As the number of Internet users increases, the public is actively commenting on their policy news stories. Current policy polls tend to rely heavily on phone and offline surveys. Collecting and analyzing policy articles is useful in policy surveys. In this study, we propose a method of analyzing comments using deep learning technology showing outstanding performance in various fields. In particular, we designed various models based on the recurrent neural network (RNN) which is suitable for sequential data and compared the performance with the support vector machine (SVM), which is a traditional machine learning model. For all test sets, the SVM model show an accuracy of 0.73 and the RNN model have an accuracy of 0.83.

▶ Keyword: recurrent neural network, opinion mining, semantic analysis

1. Introduction

한국인터넷진흥원(KISA)의 2015년 인터넷 이용 실태 조사에 따르면 인터넷 이용자 수가 4,300만 명을 돌파했고, 스마트폰 보급률 또한 85%가 넘었다 [1]. 인터넷 및 스마트폰의 발달로 인해 온라인 뉴스의 소비가 인쇄 신문을 뛰어 넘었고, 대중들이 댓글을 통해 자신의 의견을 자유롭게 표출 할 수 있게 되었다.

정치 분야 뉴스에서 각종 언론사에서는 정책에 대한 중요 사실을 포함한 기사를 실시간으로 업로드 하고 있으며, 대중들은 정부 정책과 관련된 뉴스 기사에 적극적으로 자신의 의견을 표출한다.

국가에서 정책을 추진할 때, 정책이 어느 정도의 지지를 받고 있는지를 조사하는 것이 의사결정을 하는데 중요한 역할을 할 수 있다. 현재의 정책 여론 조사 방법은 전화와 오프라인 설문 조사에 깊게 의존하는 경향이 큰데, 정책 기사 댓글을 수집하고 분석해 정책에 대한 여론을 분석 할 수 있다면 정책 여론 조사에 많은 도움

이 될 것이다.

기사 댓글을 분석하고 활용하려는 시도는 이전에도 많았지만 대부분 감정 사전을 이용하거나 고전적인 기계학습 모델로 분석하려는 시도에 그쳤으며, 상품 평판 조사 혹은 악성 댓글 탐지가 주된 연구 목표였다 [2, 3].

한편, 최근 인공지능 분야에서 딥러닝이 많은 주목을 받고 있다. 딥러닝은 깊은 구조로 이루어진 인공 신경망의 학습 기법을 일컫는다. 딥러닝은 이미지, 음성, 자연어처리 등의 다양한 영역에서 뛰어난 성과를 거두며 지속적인 관심을 받고 있다 [4, 5, 6].

본 연구에서는 딥러닝 기술을 기반으로 정책에 대한 댓글 분석 모델을 설계하고, 실험을 통해 성능을 비교 분석 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 댓글을 분석한 다른 연구 결과를 살펴보고, 본 실험에 적용된 딥러닝 기술의 이론적 배경을 설명한다. 3장에서는 본

*First Author: Dae-Kyung Sung, Corresponding Author: Young-Seob Jeong

*Dae-Kyung Sung (kdcal035@knu.ac.kr), Dept. of Computer Science and Engineering, Kyungpook National University

**Young-Seob Jeong (bytecell@sch.ac.kr), Dept. of Big Data Engineering, Soonchunhyang University

Received: 2017. 11. 01, Revised: 2017. 12. 13, Accepted: 2018. 01. 02.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP; Ministry of Science, ICT & Future Planning) (No. 2017017836). This work was supported by the Soonchunhyang University Research Fund.

연구에서 제시한 시스템의 전체적인 구조에 대해서 소개한다. 4장에서는 실험에 사용한 데이터, 학습 파라미터, 실험결과에 대해 다루고 5장에서는 결론 및 향후 연구 방향을 제시한다.

II. Preliminaries

1. Related works

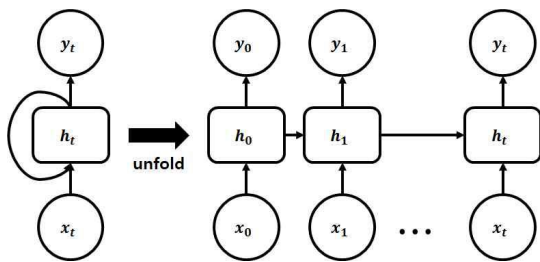


Fig. 1. Basic structure of RNN

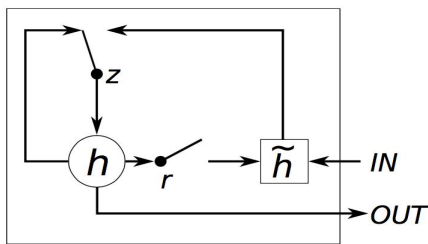


Fig. 2. GRU Gating

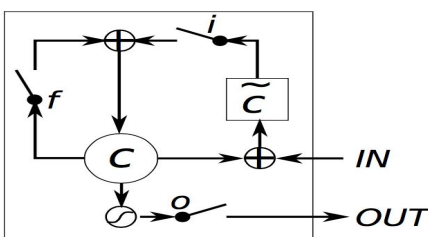


Fig. 3. LSTM Gating

온라인 뉴스기사 댓글 분석 관련 연구에서 주로 사용된 기법은 사전 기반의 감정 분석 기술이다 [7, 8]. 감정 사전 기반 감정 분석 모델은 긍정적인 의미를 띄는 단어와 부정적인 의미를 띄는 단어를 구별하기 위한 감정 사전을 구축해 문장의 긍정적인 혹은 부정적인 정도를 추출해 내는 방법이다. 학습이 필요 없고 계산 속도가 빠르다는 장점이 있지만, 비교적 정확도가 떨어진다.

Support Vector Machine(SVM) 모델 또한 댓글 분석 연구에 자주 활용 되어왔다 [9]. SVM 모델은 두 개의 범주로 구분된 학습 데이터를 구분하는 분리경계면

(Hyperplane) 중 두 그룹 간의 마진(Margin)을 최대로 하는 분리경계면을 학습을 통해 찾아내고, 이 분리경계면을 이용해 새로운 입력이 어느 범주에 속하는지 판단하는 모델이다. 이 모델은 다른 고전적인 기계 학습 모델에 비교하여 우수한 성능을 보였다[11].

딥러닝을 활용해 뉴스 댓글에 있는 대중들의 정책에 대한 의견을 분석하려는 시도는 찾아 볼 수 없었지만, 소셜 미디어에서 특정 대상에 대한 평판을 분석하는 오피니언 마이닝 분야에서 기존에 사용되던 방법들보다 딥러닝 기술이 더 잘 동작한다는 연구가 발표 되었다 [12, 13]. 본 연구에서는 이러한 딥러닝 기술을 정책 기사 댓글 분석에 활용 하였다.

자연어의 각 단어 하나의 의미만으로는 문장 전체 의미를 파악하기 어렵다. 전체 의미 파악을 위해서는 단어의 앞뒤 문맥을 고려해야 하는데, 연속된 단어들을 고려함으로써 문맥정보를 활용하기 위해 고안된 딥러닝 기법이 Recurrent Neural Network (RNN) 다. RNN은 다양한 자연어처리 문제에 대해 뛰어난 성능을 보이고 있다 [14, 15, 16].

Fig 1은 RNN의 기본 구조를 나타낸 것이고, 이를 이용해 댓글을 구성하는 각각의 단어들을 벡터로 만들어 학습을 진행한다. RNN은 일반적으로 문장이 길어질 경우 성능이 저하 되는데, 그 원인은 학습의 과정에서 출력단의 오차가 충분히 전달되지 못하기 때문이다. 이 문제를 극복하고자 하는 다양한 구조의 RNN 모델들이 제안 되었으며, 대표적인 모델로 Long Short-Term Memory (LSTM)과 Gated Recurrent Unit(GRU)가 있다 [17, 18]. Fig. 2, Fig. 3은 각각 GRU 게이트와 LSTM 게이트의 구조를 나타낸 것이다. LSTM과 GRU는 은닉층의 노드의 상태 값들이 입력되거나 출력되는 곳에 특수한 게이트를 설치하여 출력단 오차가 충분히 전달 될 수 있게 하였다 [19].

본 연구에서는 위 RNN 기반 모델들을 이용해 댓글을 분석하고 다른 모델과 성능을 비교 분석하였다.

III. The Proposed Scheme

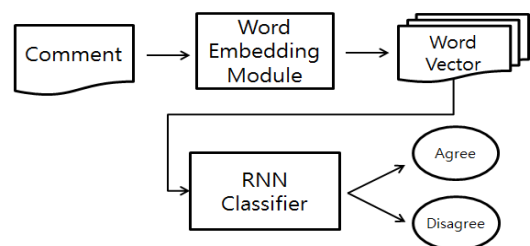


Fig. 4. System Structure

Fig. 4는 본 연구에서 제시하는 시스템의 전체적인 구조를 나타낸 것이다. 댓글을 가져온 후에 이를 워드 임베딩 모듈에서 단어 벡터의 나열로 나타내는 과정을 거친 후, 분류기에서 정책에 찬성하는지 반대하는지를 분류한다.

1. Word Embedding Module

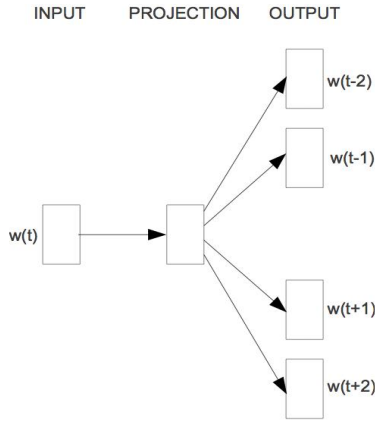


Fig. 5. skip-gram

워드 임베딩 모듈에서는 단어들을 임의의 차원의 벡터 공간에 배치하는 워드 임베딩을 수행 한다. 워드 임베딩을 하기 위해서는 사전에 학습 과정을 거쳐야 하는데, 이때 필요한 문장은 수집한 댓글과 기사 내용을 사용하였으며, Fig. 5와 같이 등장하는 단어 하나로 주변에 등장하는 단어들을 유추하고 이를 통해 학습하는 skip-gram 방식으로 학습을 진행 하였다[20]. 한국어 문장에는 여러 가지 조사와 결합해 같은 어근을 가진 다양한 형태의 단어가 존재하고 다른 의미임에도 같은 글자로 구성된 동음이의어도 존재한다. 이들을 잘 처리하기 위해서 댓글에 있는 단어를 그대로 사용하지 않고 형태소 분석과 품사 태깅을 한 후, 형태소와 품사의 쌍을 하나의 단어로 취급하였다.

학습된 워드 임베딩 모듈에서는 입력으로 들어온 댓글을 단어 벡터의 나열로 표현한다. 학습 과정과 동일하게 형태소와 품사의 쌍으로 만든 후, 벡터화를 차례로 수행한다. i 번째 댓글 $C_i = (W_1, W_2, \dots, W_m)$ 가 m 개의 단어 $W_j (1 \leq j \leq m)$ 들로 이루어져 있을 때, 형태소 분석기는 C_i 를 입력으로 받고 형태소 분석 결과물인 $M_i = (P_1, P_2, \dots, P_n)$ 를 산출한다. M_i 를 구성하는 각 요소 $P_k = \{m_k, pos_k\} (1 \leq k \leq n)$ 는 형태소 m_k 와 해당하는 품사 태그 pos_k 로 구성된다. P_k 는 d 차원 벡터로 임베딩 되어, 최종적으로 댓글 C_i 는 행렬 $V_i \in \mathbb{R}^{n \times d}$ 로 표현 된다.

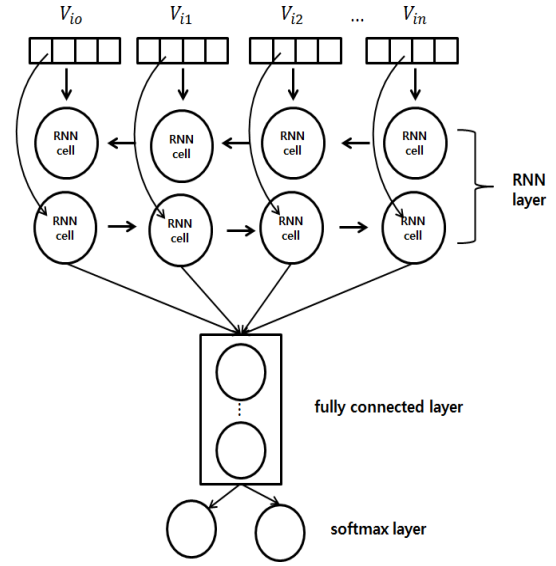


Fig. 6. Classifier structure

2. RNN based Classifier

Fig. 6는 RNN 기반 분류기 구조를 나타낸 것이다. 워드 임베딩 모듈에서 전달된 n 개의 형태소 벡터 $V_{i0}, V_{i1}, \dots, V_{in}$ 을 RNN layer로 전달한다. RNN layer는 시간 스텝 t 에 대해 s 차원 벡터인 $h_t \in \mathbb{R}^s$ 를 출력하는 RNN cell로 구성되어 있으며, RNN cell로 LSTM과 GRU를 적용하였다. 정방향과 역방향으로 진행되는 layer 각각에서 출력하는 값을 평균을 취해 사용하는 Bidirectional RNN을 사용하였다 [21]. RNN layer를 거친 후에 n 개의 RNN cell 출력 값 들을 하나의 벡터 $x_t \in \mathbb{R}^{h \times n}$ 로 만들어 fully connected layer로 전달해 차원을 축소한다. 최종적으로 softmax layer에서 댓글이 정책에 대해서 찬성할 확률과 반대할 확률로 표현 한다.

2.1 RNN Layer - LSTM

LSTM기반 RNN layer에서는 댓글 행렬 V_i 에 대해 t 가 현재 시간 스텝을 나타낼 때, hidden state의 출력 값 h_t 는 다음과 같이 계산한다.

$$i_t = \sigma(W_i \cdot [h_{t-1}, V_{i,t}] + b_i) \quad (1)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, V_{i,t}] + b_f) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, V_{i,t}] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o [h_{t-1}, V_{i,t}] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

입력 게이트 i_t 는 새 cell state 값을 계산하는데 있어

서 입력 벡터 값을 얼마나 사용할 지를 결정한다. 망각 게이트 f_t 는 이전 cell state 값을 얼마만큼 기억하고 있을지를 정해준다. 식 (3), (4)는 cell state를 계산하는 방법을 보여주는데, (3)에서는 cell state 후보 값 \tilde{C}_t 를 계산 하고, (4)에서는 이전 state C_{t-1} 에 망각 게이트 출력 값 f_t 를 곱하고 후보 값 \tilde{C}_t 에 입력 게이트 i_t 를 곱해 새로운 cell state를 생성한다. 출력 게이트 o_t 는 현재의 내부 state 값의 얼마만큼을 외부에 전달 할 지를 결정한다. 최종 출력 값 h_t 는 $\tanh(C_t)$ 에 출력 게이트 값 o_t 를 곱해 출력하며, h_1 부터 h_n 까지 결합하여 $h_{1:n} \in \mathbb{R}^{s \times n}$ 행렬로 출력된다.

2.2 RNN layer - GRU

GRU기반 RNN layer에서 출력 값 h_t 는 다음과 같이 계산한다.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (7)$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (8)$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \quad (9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (10)$$

리셋 게이트 r_t 는 새로운 입력을 이전 메모리와 어떻게 합칠 지를 정해주는데, 출력 후보 값을 계산할 때 이전 출력 값 h_{t-1} 에 리셋 게이트 출력 값을 곱함으로써 이전 출력을 리셋 게이트 출력 값 만큼 반영함을 의미한다. 업데이트 게이트 z_t 는 이전 메모리를 얼마만큼 기억할 지를 정해준다. 수식 (10)에서 업데이트 게이트 출력 값으로 과거 정보인 h_t 와 출력 후보인 \tilde{h}_t 를 어떻게 조합 할지 결정해 $h_{1:n} \in \mathbb{R}^{s \times n}$ 행렬로 출력한다.

2.3 Fully connected layer와 Softmax layer

Fully connected layer는 여러 층의 뉴럴 네트워크로 구성 되어 있으며, 본 연구에서는 L=3으로 적용하였다.

첫 layer 출력 값 h^1 과 l 번째 층 출력 값 $h^l (1 \leq l \leq L)$ 은 다음과 같이 계산 된다.

$$h^1 = f(W^1 \cdot h_{1:n} + b^1) \quad (11)$$

$$h^l = f(W^l \cdot h^{l-1} + b^l) \quad (12)$$

첫 layer에서는 RNN layer 출력 값 $h_{1:n}^m$ 을 전달 받아 해당 층의 W^1 곱하고 바이어스 b^1 값을 더한 후 활성화함수 f 를 거친 값을 다음 층으로 전달하는 연산을 수행한다. 본 연구에서 활성화함수는 ReLU 함수를 사용하였다. 두 번째 층부터 이전 층의 출력 값을 전달 받아 동일한 연산을 수행하며 차원을 줄여 나가는 수식 (12)와 같은 연산을 수행한다. 최종 출력 $h^L \in \mathbb{R}^2$ 을 Softmax layer로 전달해 정책에 대해서 찬성할 확률과 반대할 확률 값으로 표현 한다.

Table 1. Dataset

number	policy	agree	disagree	total
<1>	Phase out nuclear power plant	2127	325	2452
<2>	Dementia national responsibility	341	494	835
<3>	National university integration	630	542	1170
<4>	Greenstone belt relaxation	454	993	1447
	total	3552	2354	5906

Table 2. Data Example

number	content	label
<3>	이런 데에 돈 쓰라고 뽑은거야, 사대강 같은데에 퍼붓지 말고	agree(0)
	연합 대학 너무 너무 들어가고 싶다. 문썸때 최고의 업적	disagree(1)
<4>	그린벨트 해제 안할거면 국가가 사라. 내땅 내가 어찌 못하는 경우가 어딴나?	agree(0)
	탁상행정의 진수를 보여주려는가?? 역시 유능하십니다.	disagree(1)

Table 3. accuracy by model, where <n> denotes the dataset index in table1

model	<1>	<2>	<3>	<4>
SVM + unigram	84.5	91.6	65.1	67.9
SVM + bigram	84.6	91.6	67.2	67.7
LSTM + onehot	86.4	85.3	71.3	70.3
GRU + onehot	86.8	87.3	71.7	70.4
bidirectional LSTM + onehot	90.2	91.3	73.3	71.6
bidirectional GRU + onehot	90.4	91.3	73.4	72.3
bidirectional LSTM + word2vec	88.7	91.4	73.1	69.7
bidirectional GRU + word2vec	89.2	91.3	72.5	70.3

IV. Experiment

1. Data Set

포털 사이트에 많은 댓글이 달리는 정책을 선정해 관련 기사 댓글들을 수집했다. table 1은 수집한 정책의 종류와 댓글의 분포를 나타낸 것이다. 댓글 하나에 포함된 모든 문장들을 한 건의 데이터로 취급 했고, 네 개의 정책과 관련된 댓글 5, 906개를 수집했다. <1>은 탈원전 정책과 관련한 뉴스기사 댓글들이고, <2>는 치매 국가 책임제에 관한 댓글이다. <3>은 국립대 통합 정책과 관련한 댓글이고, <4>는 그린벨트 완화 정책 관련 댓글이다. table 2는 데이터 예시를 나타낸 표이다. 한 건의 데이터는 댓글 내용과 정책에 대해 찬성인지 반대인지를 나타내는 태그로 이루어져 있다. 태그하는 작업은 일관성 유지를 위해 한 사람이 하나의 주제에 속한 모든 댓글에 대하여 찬성은 0, 반대는 1을 기록했다.

2. Performance Analysis

table 3은 모델별 성능을 나타낸 표이다. 주제를 나타내는 <1> ~ <4>는 table 1에서 사용한 주제 번호를 사용하였다. 성능 측정 방법은 댓글이 정책에 대해 찬성인지 반대인지 여부를 모델이 예측한 정답과 사람이 직접 판단한 정답이 일치하는 비율을 나타낸 것이다. table 2 정책에 반대하는 댓글과 같이, 정책에 반대하는 댓글임에도 반어법을 사용하여 반대 댓글과 유사해지는 경우가 <3>과 <4>에 비교적 빈번하게 발생하였고, 다른 데이터에 비해 낮은 성능을 유발한 원인으로 생각된다.

SVM 모델은 scikit-learn 라이브러리를 사용했고 [22], 디폴트 파라미터인 C value 1.0과 degree 값 3의 rbf 커널 이진 분류 SVM을 사용했다. 공정한 비교를 위해 RNN 계열과 동일하게 형태소와 품사 태그를 묶어 unigram과 bigram 방식으로 댓글을 벡터로 표현해 자질로 사용했다. unigram에 비해 bigram 방식이 벡터 차원이 훨씬 컸지만, 정확도는 크게 차이가 없었다. SVM은 학습데이터 분포에 영향을 받아 한쪽으로 치우친 방향으로 결과를 예측했다. Fig. 7은 실제 데이터 분포와 SVM이 예측한 데이터 분포를 비교한 표이다. 이러한 이유로 가장 편향된 데이터를 가진 탈원전 주제에서는 SVM 모델이 가장 높은 성능을 보인 것으로 예상된다.

RNN 기반 모델은 cross-entropy를 목적 함수로 해 adam optimizer로 훈련을 시켰으며, learning rate는 0.001을 사용하고 300 epoch만큼 학습을 진행했다. One-hot 인코딩으로 표현한 형태소 벡터와 word2vec 형태소 벡터 두 가지를 자질로 사용했다. RNN 기반 모델들이 SVM 모델에 비해 비교적 높은 성능을 보였는데, SVM 모델에서는 무시되는 연속성이 RNN 기반 모델에서는 반영되기 때문인 것으로 보인다. 가장 높은 성능을 보인 것은 GRU Cell

을 사용한 bidirectional RNN이며, 양방향으로 단어의 연속성을 반영 하였기 때문에 문맥을 더 잘 파악한 것으로 보인다.

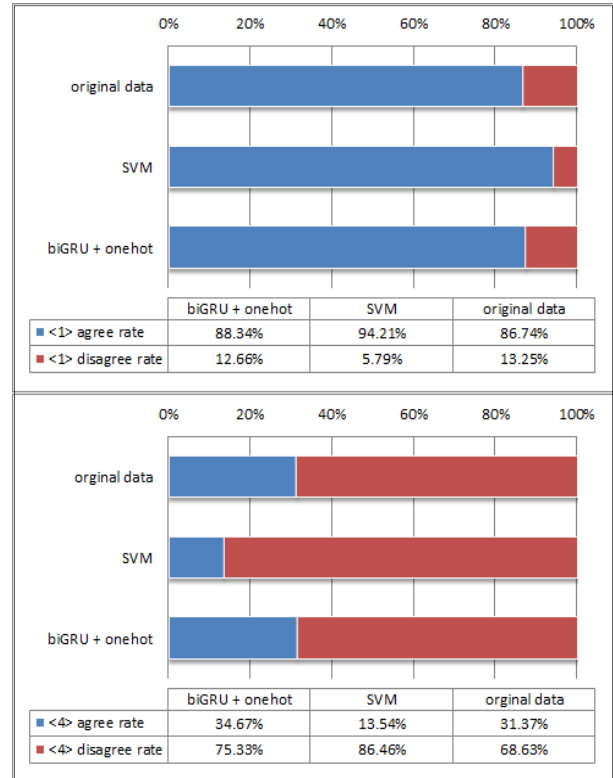


Fig. 7. Predictive distribution by model versus actual data

Word2vec을 적용하는 것은 성능 측면에서는 도움이 되지 못했지만 학습된 Word2vec 모듈을 사용 시 단어 벡터의 차원을 줄여 학습시간과 테스트 시간을 줄여주는 효과가 있었다. RNN기반 모델들이 Word2vec feature를 사용한 경우와 one-hot encoding feature를 사용한 경우의 학습 시간과 테스트 시간을 table 4에 나타냈다.

제안 된 모델에 Attention layer를 적용해 보았으나 성능 향상에 도움이 되지 못했다. Attention은 시퀀스 길이가 긴 데이터가 있는 복잡한 문제를 효과적으로 학습하기 위해 시퀀스 요소들에 가중치를 두는 학습 방법이다. 이를 적용해 복잡한 문제들에 대해 성능 향상을 시킨 연구들이 있는 반면 [23, 24, 25], 크게 효과를 보지 못한 사례도 있다 [26]. 이 논문에서는 Attention layer를 적용 할 만큼 복잡하지 않은 문제에 대해서 일반적인 Attention mechanism을 적용해 학습하는 것은 과잉이 될 수 있다 설명했다.

본 연구에서도 대부분의 댓글이 길이가 짧은 비교적 간단한 데이터를 대상으로 했기 때문에, Attention layer가 성능 향상에 크게 도움이 되지 못한 것으로 보인다.

Table 4. Learning and testing time (s)

model	train	test
LSTM + onehot	1,132	79
GRU + onehot	1,017	77
LSTM + word2vec	217	34
GRU + word2vec	205	33

V. Conclusions

본 연구에서는 RNN 기반 딥러닝 모델을 정책 뉴스 기사 댓글 분석에 이용해 방법을 제안 하였으며, 기존 댓글 분석에 주로 사용되던 SVM 모델에 비해 높은 성능을 보였다. 앞으로 정책 여론 조사 등에 유용하게 활용될 것으로 기대된다. 학습에 보다 많은 데이터를 필요로 하는 딥러닝의 특성 상 데이터를 추가 시 성능이 더 향상될 것으로 예상되며, CNN 기반 모델 등의 다른 딥러닝 기술도 시도할 예정이다.

REFERENCES

- [1] KISA. "Internet Usage Survey in 2015", KISA, 2015.
- [2] Jinju Hong, Sehan Kim, Jeawon Park & Jaehyun Choi. "A Malicious Comments Detection Technique on the Internet using Sentiment Analysis and SVM, Journal of the Korea Institute of Information and Communication Engineering, 20(2), 260~267, 2016.
- [3] Hana Cho, Yeounoh Chung, Jaedong Lee and Jee-Hyong Lee, "Sentiment Analysis Using News Comments for Public Opinion Mining," Journal of the Korea Institute of Intelligent Systems, Vol. 23, No. 1, pp. 149~150, 2013.
- [4] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553. 436-444, 2015
- [5] Dong, Chao, et al. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38.2, 295-307. 2016.
- [6] Lample, Guillaume, et al. "Neural architectures for named entity recognition." arXiv preprint arXiv:1603.01360, 2016.
- [7] Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." Proceedings of the 2nd international conference on Knowledge capture. ACM, 2003.
- [8] Jinju Hong, Sehan Kim, Jeawon Park and Jaehyun Choi, "A Malicious Comments Detection Technique on the Internet using Sentiment Analysis and SVM," Journal of the Korea Institute of Information and Communication Engineering, Vol. 20, No. 2, pp. 260~267, 2016.
- [9] Myo-Sil Kim and Seung-Shik Kang, "A Design and Implementation of Malicious Web Log Identification System by Using SVM," Proceedings of the 6th Annual Conference on Human and Cognitive Language Technology, , pp. 285~289, 2006
- [10] Zhang, Dongwen, et al. "Chinese comments sentiment classification based on word2vec and SVM perf." Expert Systems with Applications 42.4 , 2015.
- [11] Huang, Jin, Jingjing Lu, and Charles X. Ling. "Comparing naive Bayes, decision trees, and SVM with AUC and accuracy." Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, 2003.
- [12] Kalchbrenner, Nal, and Phil Blunsom. "Recurrent Continuous Translation Models." EMNLP. Vol. 3. No. 39. 2013.
- [13] Irsoy, Ozan, and Claire Cardie. "Opinion Mining with Deep Recurrent Neural Networks." EMNLP. 2014.
- [14] Yang, Zichao, et al. "Hierarchical Attention Networks for Document Classification." HLT-NAACL. 2016.
- [15] J.-P.S. Draye, D.A. Pavisic, G.A. Cheron, G.A. Libert, "Dynamic recurrent neural networks: a dynamical analysis", Systems Man and Cybernetics Part B: Cybernetics IEEE Transactions on, vol. 26, pp. 692-706, 1996
- [16] Yao, Kaisheng, et al. "Recurrent neural networks for language understanding." Interspeech. 2013.
- [17] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8, 1997.
- [18] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 2014.
- [19] Fu, Rui, Zuo Zhang, and Li Li. "Using LSTM and GRU neural network methods for traffic flow prediction." Chinese Association of Automation (YAC), Youth Academic Annual Conference of. IEEE, 2016.
- [20] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 2013.
- [21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.
- [22] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12.Oct 2825-2830, 2011.
- [23] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint

arXiv:1508.04025 2015.

- [24] Bahdanau, Dzmitry, et al. "End-to-end attention-based large vocabulary speech recognition." Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.
- [25] Yang, Zichao, et al. "Stacked attention networks for image question answering." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. APA, 2016.
- [26] Aharoni, Roei, Yoav Goldberg, and Israel Ramat-Gan. "Morphological Inflection Generation with Hard Monotonic Attention." Proceedings of ACL, 2017

Authors



Dae-Kyung Sung is currently a B.S. candidate in the Department of Computer Science and Engineering at Kyungpook National University. His research interests include deep learning, text mining and big data analysis.



Young-Seob Jeong received a BS in Computer Science from Hanyang University, Korea, in 2012, an MSc in Computer Science from KAIST, Korea, and in 2016, a PhD in School of Computing from KAIST, Korea. He joined the faculty of the Department of Big Data Engineering at Soonchunhyang University, Asan city, Korea, in 2017. His current research topics are text mining, information extraction, action recognition, and dialog systems, where his favorite techniques are topic modeling and deep learning.