

# CNN-LSTM을 이용한 대화 문맥 반영과 감정 분류

신동원<sup>†,††</sup>, 이연수<sup>††</sup>, 장정선<sup>††</sup>, 임해창<sup>†</sup>

고려대학교<sup>†</sup>, {yeonsoo, jsjang}@ncsoft<sup>††</sup>

littmus@korea.ac.kr, {yeonsoo, jsjang}@ncsoft.com, rim206@gmail.com

## Using CNN-LSTM for Effective Application of Dialogue Context to Emotion Classification

Dong-Won Shin<sup>†,††</sup>, Yeon-Soo Lee<sup>††</sup>, Jung-Sun Jang<sup>††</sup>, Hae-Chang Rim<sup>†</sup>  
Korea University<sup>†</sup>, NCSoft Corp.<sup>††</sup>

### 요약

대화 시스템에서 사용자가 나타내는 발화에 내재된 감정을 분류하는 것은, 시스템이 적절한 응답과 서비스를 제공하는데 있어 매우 중요하다. 본 연구에서는 대화 내 감정 분류를 하는데 있어 직접적, 간접적으로 드러나는 감정 자질을 자동으로 학습하고 감정이 지속되는 대화 문맥을 효과적으로 반영하기 위해 CNN-LSTM 방식의 딥 뉴럴 네트워크 구조를 제안한다. 그리고 대량의 구어체 코퍼스를 이용한 사전 학습으로 데이터 부족 문제를 완화하였다. 실험 결과 제안하는 방법이 기존의 SVM이나, 단순한 RNN, CNN 네트워크 구조에 비해 전반적인 성능 향상을 보였고, 특히 감정이 있는 경우 더 잘 분류하는 것을 확인할 수 있었다.

주제어: 감정 분류, 딥 뉴럴 네트워크, RNN, CNN, LSTM

### 1. 서론

본 연구는 대화 속에서 나타나는 사용자의 감정을 분류하고자 한다. 시스템은 사용자의 감정을 인식함으로써, 보다 적절한 응답과 서비스를 제공할 수 있다. 최근 Affective Computing과 관련된 연구 결과에 따르면 사용자의 감정을 인지하고 이에 따라 에이전트가 적극적인 대응을 할 때 사용자도 시스템을 더욱 효과적으로 이용할 수 있게 된다고 한다[1].

본 연구에서는 Plutchick의 8분류 체계[2]인 기쁨(Joy), 신뢰(Trust), 두려움(Fear), 놀람(Surprise), 슬픔(Sadness), 혐오(Disgust), 화남(Anger), 기대(Anticipation)에 감정 없음(None), 미안함(Sorryness), 부러움(Enviousness)의 추가적인 감정 카테고리를 사용하여 대화 내 발화들이 나타내는 다양한 감정을 분류하고자 한다. 기존의 감정 분류 연구들은 대부분 대화가 아닌 일반 텍스트에서 감정 분류(Sentiment Classification)와 같은 극성 분류에 초점을 맞추었다. 긍정/부정 외에 기쁨, 슬픔, 두려움 등 다양한 감정 분류에 관한 연구는 많지 않으며, 특히 대화적 특징을 반영한 대화 속 감정 분류 연구 역시 매우 드물다.

대화는 일반 텍스트와 다르게 화자와 청자 간에 발화를 주고받으며 이야기가 진행된다. 대화 참여자들의 감정은 다른 사람의 말에 의해 자신의 감정이 바뀌기도 하고, 한번 발생된 감정이 일정 시간 지속되기도 한다. 그리고 감정을 직접적인 어휘를 통해 드러내기도 하지만, 대화 내용을 통해 간접적으로 나타나기도 한다. 예를 들어 그림 1과 같은 경우는 보다 직접적인 어휘에 의해 하나의 발화 내에 감정이 드러난 경우이지만, 그림 2와 같

은 경우는 보다 간접적인 정황을 통해 드러나며, 한번 나타난 감정이 지속될 수 있는 상황을 보여준다.

따라서 대화 속 감정 분류 시스템은 다양한 감정 어휘 뿐 아니라, 대화 문맥 속에서 나타나는 분위기나 주제 등 감정을 유발하는 간접적인 자질들이 반영되어야 하며, 대화의 현재 발화 뿐 아니라 이전 발화들을 효과적으로 반영해야 한다.

그림 1 감정적인 대화 예시 1

“A: 이거 진짜 재밌다.” (기쁨)  
“B: 나 지금 정말 속상해.” (슬픔)

그림 2 감정적인 대화 예시 2

“A: 어제 혼났다며?”  
“B: 휴... 담임 쌤한테 맨날 지각한다고 혼났지” (슬픔)  
“A: 헐.. 많이 혼났어?”  
“B: 어” (슬픔)

본 연구에서는 이러한 대화적 특징과 간접적인 감정적 상황을 감정 분류에 반영하기 위해 CNN-LSTM (Convolutional Neural Network - Long Short Term Memory) 모델을 사용한다. CNN을 통해 입력 문장의 자질을 자동으로 학습 시킨 뒤 LSTM[3]을 통해 문맥이 반영되도록 한다. 또한 적은양의 학습 데이터로부터 효과적으로 딥 네트워크 구조를 학습하기 위해 대량의 구어체 코퍼스로부터 입력 벡터를 사전 훈련(Pre-training)시키는 방법을 제안한다.

## 2. 관련 연구

텍스트 도메인에서의 감정 분류는 대부분 텍스트의 감정 어휘를 자질로 하여 SVM과 같은 분류모델을 사용한다. [4]는 블로그 데이터에 대해 Ekman의 감정 분류체계에 따라 6가지 감정을 기준으로 SVM을 이용하여 분류하였다. [5]의 연구에서는 트위터의 트윗을 대상으로 대량의 Unigram 자질을 사용하여 감정을 분류한다. [6]과 [7]에서는 대화에서의 감정 분류가 문맥에 의존할 수 있음을 지적하고 이를 위해, 발화 내 어휘 자질 뿐 아니라 대화 문맥을 이용하였다. 그러나 문맥을 확장할수록 과도한 어휘 자질에 의존할 수밖에 없고, 대용량의 학습데이터가 없는 한 데이터 부족(Data sparseness) 문제가 심각하게 대두된다. [8]은 이러한 데이터 부족 문제를 해결하기 위해 대량의 원시 코퍼스를 사용한 Word Embedding 모델을 감정 분류 자질에 활용하는 방법을 제안하였다. 이를 통해 문제가 어느 정도 해소 되었다고 하나, 분류 모델 내에서 구조적으로 대화 히스토리를 반영한 것이 아니기 때문에 여전히 대화 문맥을 포함한 자질 튜닝(Feature Engineering)과정을 여러 번 거쳐야 했다.

최근 [9]의 CNN 모델은 단순한 Convolution과 Pooling 과정만으로 문장 및 문서 분류에서 좋은 성능을 보였다. 또한 [3]의 LSTM 모델은 시퀀셜한 입력 정보에 대해 순서를 고려한 자질의 학습과 생성 측면에서 기계번역 영역에서 뿐 아니라 여러 문제에서 좋은 성능을 보여주고 있다. [10]의 모델은 텍스트 중에서도 특히 길이가 짧고, 연속적인(주로 문서 속 문장들, 대화 속 발화와 같은) 텍스트를 RNN(Recurrent Neural Network)과 CNN을 사용해 표현하고 이를 분류에 사용하여 화행(Dialogue act) 분류에서 state-of-the-art 성능을 보여주었다. 제안하는 방법은 이와 유사하게 분류 모델 내에서 대화 히스토리를 반영하고 대량의 어휘 자질을 자동으로 학습할 수 있도록 대화 감정 분류에 CNN-LSTM 모델을 사용하는 것이다.

## 3. Deep Neural Network를 이용한 감정 분류 모델

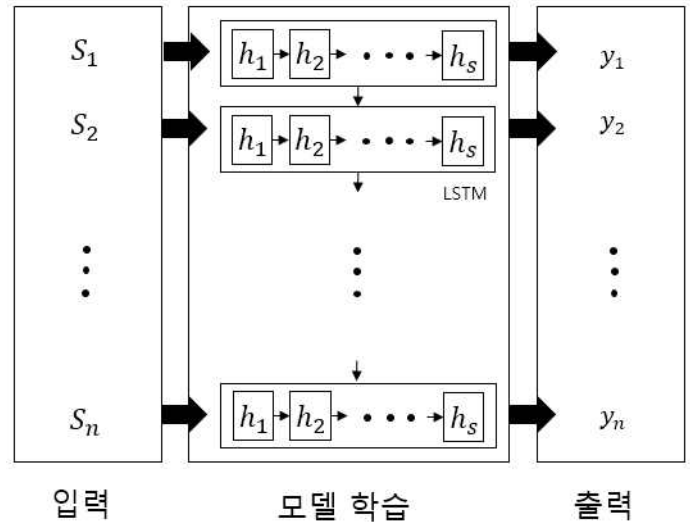
3장에서는 기존에 자연어처리 영역에서 좋은 성능을 보인 RNN과 CNN을 분류 모델로 사용해 감정 분류를 하는 방법과 모델 상의 한계점을 살펴보고, 대화에서의 문맥과 자질을 충분히 반영하기 위해 CNN-LSTM 모델을 사용하는 방법에 대해 자세히 살펴보도록 한다.

### 3.1 RNN을 이용한 감정 분류 모델

입력으로 발화의 텍스트가 들어오게 되면, 이를 RNN 레이어의 입력으로 사용하기 위해 단어 단위의 벡터로 변환하게 된다. 입력 텍스트를 구성하는 변환된 단어 벡터들은 순차적으로 RNN의 Hidden layer에 입력되어 모델 학습이 이루어진다. RNN Hidden layer에서 매 스텝의 상태 값을 결정하는 데에는 이전 상태 값과 현재의 입력을 모두 고려하게 된다. 또한 이전 문장의 출력 정보 역시

현재 문장의 학습 및 분류에 반영되는 구조이다. 이러한 과정들을 그림 3으로 나타내었다.

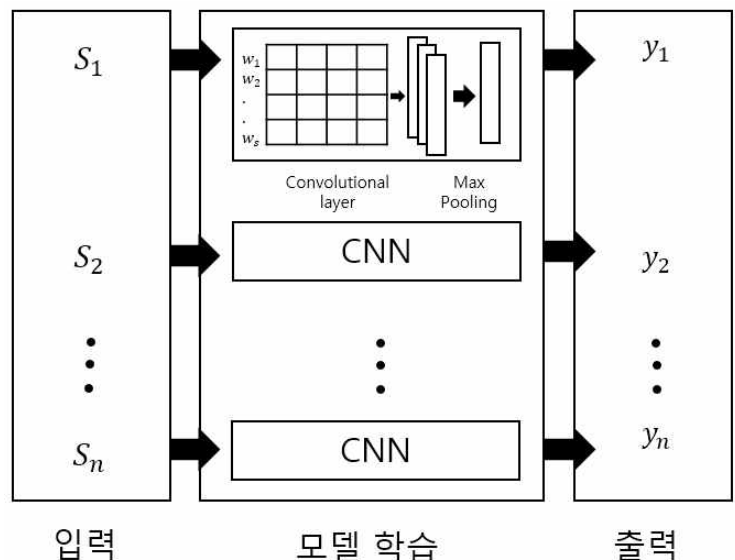
그림 3 RNN을 이용한 분류 모델



### 3.2 CNN을 이용한 감정 분류 모델

CNN 모델에서도 마찬가지로, 우선 입력된 발화의 텍스트를 벡터로 변환해 이를 Convolutional layer에 입력한다. Convolutional layer를 거치면서 입력된 텍스트를 잘 나타내는 특징들이 추출되고, 이렇게 추출된 특징들은 마지막으로 Pooling layer를 거쳐 중요한 정보만을 남긴 후 다음 레이어에 들어갈 입력을 출력하게 된다. Pooling 방법에는 Mean Pooling, Average Pooling 등 다양한 방법들이 있지만, 일반적으로 분류 문제에서는 Max Pooling이 사용된다. 이러한 과정들을 그림 4로 나타내었다.

그림 4 CNN을 이용한 분류 모델

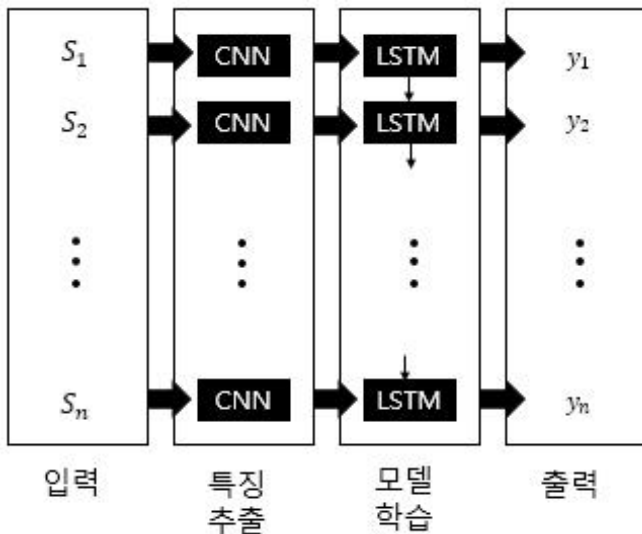


### 3.3 CNN-LSTM 모델을 이용한 감정 분류

앞서 설명한 두 뉴럴 네트워크 모델은 텍스트 분류 문제에 있어 각각의 단점을 가지고 있다. RNN 모델의 경우, 분류에 필요한 입력 텍스트의 어휘적인 특징들을 충분히 포착하기 어렵다는 단점이 있으며, CNN 모델의 경우 한 번에 한 발화의 텍스트를 입력으로 받기 때문에, 대화에서 현재 발화를 분류하는데 활용될 수 있는 이전 발화들의 정보와 같은 Long-term dependency를 구조적으로 반영하기 어렵다는 문제점이 있다.

이러한 각 모델의 단점을 서로 상호 보완하기 위해, 두 종류의 뉴럴 네트워크를 함께 활용하는 CNN-LSTM 모델을 대화에서의 감정 분류에 사용하였다. CNN 모델을 통해 입력 텍스트의 특징들을 자질 벡터로 추출하고, 이를 LSTM 레이어의 입력으로 하여 분류 모델을 학습시킨다. 그림 5는 이러한 과정을 그림으로 나타낸 것이다.

그림 5 CNN-LSTM을 이용한 분류 모델



## 4. 실험 및 평가

### 4.1 대화 코퍼스 구축 및 감정 정답 집합 구축

대화 코퍼스는 영어 회화의 한국어 번역 데이터와, 영어 드라마의 한국어 자막 데이터를 재가공하여 구축하였다. 자막 데이터에서는 여러 명의 화자가 아닌 1:1로 대화가 이루어지는 부분들만을 선택하여, 데이터를 구성하였다.

대화 코퍼스의 감정을 태깅하는 작업은 총 3명의 작업자에 의해 수행되었다. 대화의 발화에 대해 각 작업자가 감정 태깅한 결과를 다수결을 통해서 1차 감정과 2차 감정을 결정하였다. 3명의 태깅 결과가 모두 일치하지 않은 전체 발화의 5.9%에 대해서는 다시 감정을 태깅하도록 하였다. 1차 감정에 대한 작업자 3명 모두의 동의율은 48.3%로 낮게 나타났지만, 작업자 중 최소 2명까지의 동의율은 92.1%로 매우 높게 나타났다. 또한 2차 감정까지 고려했을 때 작업자 2명까지의 동의율은 94.1%로 나타났다. 작업자들의 불일치가 주로 일어난 감정 카테고리

리들은 ‘슬픔’, ‘분노’, ‘싫어함’이었다. 이러한 과정을 통해 총 2,587개의 대화, 22,005개의 발화로 대화 코퍼스를 구축하였다. 1차 감정에 대한 감정 카테고리별 발화 비율은 표 2와 같다.

표 2 감정 카테고리 별 발화 비율

감정 카테고리	발화 수	비율 (%)
감정 없음	15171	68.9
기쁨	1644	7.5
놀람	1463	6.6
화남	855	3.9
두려움	738	3.4
슬픔	725	3.3
싫어함	519	2.4
바람	504	2.3
미안함	272	1.2
부러움	67	0.3
신뢰	47	0.2
전체	22005	

일반적으로 텍스트를 벡터로 변환하기 위해 사용되는 임베딩 레이어에서는 벡터 초기화를 위해 모든 벡터 값을 0으로 하거나, Xavier Initialization[12] 과 같은 초기화 기법을 사용하게 된다. 하지만 본 연구에서는 Word2Vec[11]을 사용해 구축한 임베딩 모델을 이용해 딥 뉴럴 네트워크 모델의 입력으로 사용되는 임베딩 벡터의 초기화에 사용한다. 이를 위한 임베딩 모델의 학습에 사용되는 데이터는 표 3과 같다.

표 3 임베딩 코퍼스 데이터

코퍼스	크기	문장 수
한국어 뉴스 기사 코퍼스	2.069 GB	11,835,341
해외 드라마 자막 한국어 번역 코퍼스	18.3 MB	361,480
전체	2.087GB	12,196,821

임베딩 모델 구축에 사용되는 알고리즘인 Word2Vec에 사용되는 주요 파라미터들은 표 4와 같다. 각 파라미터들의 최적 값은 실험적으로 결정하였다.

표 4 Word2Vec 파라미터

파라미터	값
Architecture	skip-gram
Window size	9 [5-13]
Dimensionality	300 [100-500]
Training Algorithm	hierarchical softmax

Min-count	10 [5-15]
-----------	-----------

### 4.3 실험 설계 및 평가 척도

본 연구에서는 대화 코퍼스를 대상으로 하여 딥 뉴럴 네트워크 모델을 사용해 감정 분류기를 학습시켰다. 또한 제안한 모델의 성능을 비교하기 위하여 [8]의 SVM과 임베딩 자질을 사용한 감정 분류 모델을 베이스라인으로 사용하였다. 각 딥 뉴럴 네트워크 모델에 사용된 파라미터들은 표 5와 같다. 각 파라미터들의 값은 실험적으로 결정하였다.

표 5 딥 뉴럴 네트워크 모델 파라미터

모델	파라미터	값
	Number of epoch	10000
	Learning rate	$10^{-5}$ [ $10^{-4}$ – $10^{-5}$ ]
	Dropout rate	0.5 [0-1]
RNN	Output dimension	300 [100-500]
CNN	Number of filters	500 [100-500]
	Filter height	4 [2-5]

딥 뉴럴 네트워크 모델을 학습하는 데에는 많은 시간이 소요되므로, 모델의 분류 성능을 검증하는데 있어 교차 검증을 사용하기 어렵다는 문제가 있다. 때문에 대화 코퍼스에서 ‘감정 없음’ 카테고리들과 나머지 감정 카테고리들의 비율을 최대한 유사하게 하도록 대화들을 선택하여, 학습/검증/테스트 데이터를 각각 대화 2000개 (발화 17062개), 293개 (2512개), 294개 (2431개)로 구성하였다.

모델의 감정 분류 성능을 평가하기 위한 평가 척도는 다음과 같다. 앞서 대화 코퍼스의 감정 태깅에서 살펴본 바와 같이, 대화 속에서 사용자들이 나타내는 발화가 어떤 감정을 내재하는지에 대해서는 모호성이 다수 존재한다. 하지만 감정 태깅 결과 1,2차 감정 중 한 감정에 대해서는 작업자들의 동의율이 높게 나타난다는 점을 반영하여 분류 대상 발화에 정답이 2차까지 있는 경우에는, 분류 모델이 두 정답 중 하나를 맞추면 맞은 것으로 하는 평가 척도를 사용 한다.

$$Accuracy = \frac{1,2차정답 중 하나와 모델출력이 일치한 수}{전체 발화 개수}$$

### 4.3 실험 결과 및 분석

딥 뉴럴 네트워크 모델의 사용이 대화에서의 감정 분류에 있어 어떤 성능을 나타내는지 평가하기 위하여, 베이스라인 모델과 RNN, CNN, CNN-LSTM 모델의 성능을 비교하였다.

표 6을 보면, 전체적인 감정 분류 성능은  $RNN < SVM < CNN$  순으로 나타나며, RNN과 CNN을 함께 사용한

CNN-LSTM 모델이 가장 좋은 성능을 나타낸 것을 확인할 수 있다.

표 6 분류 모델 별 감정 분류 성능

분류 모델	Accuracy (%)
SVM	81.69
RNN	81.04
CNN	82.35
CNN-LSTM	<b>82.93</b>

분류 성능을 좀 더 자세히 분석하기 위하여, 각 분류 모델의 감정 카테고리 별로 정답을 얼마나 맞혔는지를 살펴보았다. 표 7을 보면, ‘감정 없음’ 카테고리에서는 기존의 SVM 모델이 가장 많이 정답을 맞혔지만, 나머지 감정 카테고리들에 대해선 딥 뉴럴 네트워크 모델들이 SVM 모델보다 더 많이 맞춘 것을 확인할 수 있다.

SVM 모델에서 ‘감정 없음’ 카테고리를 가장 많이 맞춘 것은 학습 데이터에 대한 과적합(Over-fitting) 때문이라고 해석할 수 있다. 반면에 딥 뉴럴 네트워크 모델들은 Dropout과 같은 네트워크가 학습 데이터에 과적합 되는 것을 방지할 수 있는 기법을 적용할 수 있어, ‘감정 없음’ 카테고리의 분류 성능은 다소 감소하지만, 감정이 나타나는 발화들의 감정 분류 성능은 베이스라인에 비해 상당히 향상된 것으로 보인다. 다만 ‘부러움’, ‘신뢰’와 같이 감정을 나타내는 발화들의 수가 충분히 많지 않은 발화들을 분류하는 데에 있어서는 모든 모델들이 정답을 거의 맞히지 못해 좋지 않은 성능을 보였다.

표 7 감정 카테고리 별 모델의 분류 성능

감정 카테고리	정답 개수			
	SVM	RNN	CNN	CNN-LSTM
감정 없음	<b>1835</b>	1772	1821	1824
기쁨	70	<b>88</b>	76	81
놀람	26	<b>27</b>	<b>33</b>	32
화남	5	<b>11</b>	8	9
두려움	1	<b>4</b>	2	4
슬픔	35	<b>44</b>	43	39
싫어함	0	<b>2</b>	0	<b>2</b>
바람	5	8	4	<b>10</b>
미안함	9	14	<b>15</b>	<b>15</b>
부러움	0	0	0	0
신뢰	0	0	0	0
총합	1986	1970	2002	<b>2016</b>

다음으로 표 8을 보면, RNN 모델과 CNN 모델을 비교했을 때 RNN 모델에서는 ‘감정 없음’ 카테고리의 분류 성능이 상당히 떨어졌지만, 나머지 감정 카테고리들을 가장 많이 맞춘 것을 확인할 수 있다. 반면에 CNN 모델에

서는 ‘감정 없음’의 분류 성능을 높이면서, 동시에 감정이 있는 발화들에 대한 분류 성능도 베이스라인에 비해 향상되어 전체적으로 좋은 감정 분류 성능을 보였다.

또한 두 딥 뉴럴 네트워크 모델들의 단점을 상호 보완한 CNN-LSTM 모델은 RNN 모델보다 ‘감정 없음’의 분류 성능을 높이면서, 동시에 CNN 모델보다 감정이 있는 발화들의 분류 성능까지 높여 전체적인 분류 성능을 향상시켰다. 그 결과 실험에 사용된 모델들 중 가장 좋은 분류 성능을 보였다. CNN-LSTM 모델이 문장으로부터 감정 분류에 효과적인 특징들을 추출해 활용하며, 동시에 대화에서의 문맥을 효과적으로 고려하여 대화에서의 감정 분류 문제에 있어 효과적인 모델임을 확인할 수 있었다.

표 8 분류 모델 별 감정 없음/감정 분류 정답 개수

분류 모델	감정 없음	감정
SVM	1835	151
RNN	1772	198
CNN	1821	181
CNN-LSTM	1824	192

마지막으로 CNN-LSTM 모델에서 성능이 얼마나 향상되었는지를 더 자세히 분석하기 위하여 각 감정 카테고리들의 F1 Score를 계산하여 베이스라인인 SVM 모델과 비교하였다. 비교에는 각 모델에서 정밀도나 재현율을 계산할 수 없어 비교할 수 없는 ‘두려움’, ‘싫어함’, ‘부러움’, ‘신뢰’를 제외한 7개의 감정 카테고리만을 반영하였다. 또한 감정 카테고리에서 ‘감정 없음’의 비율이 과반수를 넘게 차지하기 때문에 이를 포함한 경우와 포함하지 않은 경우 모두에 대해 F1 Score를 계산하였다.

표 9를 보면, ‘감정 없음’을 제외했을 때의 ‘화남’, ‘슬픔’ 카테고리를 제외한 모든 경우에서 F1 Score가 상승하여 CNN-LSTM 모델의 사용이 감정 분류에 효과적임을 확인할 수 있었다.

표 9 SVM 모델과 CNN-LSTM 모델의 F1 Score

감정 카테고리	F1 Score (%)		F1 Score (%) (감정 없음 제외)	
	SVM	CNN-LSTM	SVM	CNN-LSTM
감정 없음	84.58	85.40		
기쁨	51.74	61.83	88.16	94.19
놀람	27.32	33.33	75.76	84.21
화남	8.76	11.27	66.67	64.00
슬픔	47.41	48.57	84.21	81.93
바람	11.59	25.97	66.67	80.00
미안함	51.43	71.43	94.74	100.00
Macro	75.97	77.56	83.38	87.59
Micro	40.40	48.26	79.37	84.05

## 5. 결론

본 연구는 딥 뉴럴 네트워크 모델을 활용해 대화에서의 감정 분류 성능을 향상시켰다. 특히 감정이 대화에서 나타나는 양상을 반영하여 문맥을 고려할 수 있는 효과적인 네트워크 구조를 적용하고 다양한 모델과 비교 분석하였다. 또한 임베딩된 입력을 통해 데이터 부족 문제를 완화하고 Dropout을 통해 과적합 문제를 최소화하도록 하였다. 실험 결과 “감정 없음” 라벨의 분류 성능은 다소 감소하지만, 이외 감정들에 대한 분류 성능이 전반적으로 향상됨을 보였으며, CNN-LSTM 모델이 대화 문맥을 반영한 감정 분류에 있어 기존의 SVM 모델이나 단순 RNN, CNN을 사용한 모델에 비해 적합하다는 것을 알 수 있었다. 향후에는 RNN에서 Attention Mechanism을 적용하여, 문장의 어떤 부분, 대화에서 어떤 문장이 분류에 영향을 미치는지 분석할 수 있는 여지가 있고, 딥 뉴럴 네트워크 구조의 개선 및 파라미터 최적화를 통해 성능을 더 향상시킬 필요가 있다.

## 사사

이 논문은 2012년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2012M3C4A7033344)

## 참고문헌

- [1] Picard, Rosalind W., and Roalind Picard. Affective computing. Vol. 252. Cambridge: MIT press, 1997.
- [2] Robert Plutchik, “A general psychoevolutionary theory of emotion”, In Emotion: Theory, Research, and Experience: Vol. 1. Theories of Emotion, pp. 3-33, 1980
- [3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation, pp. 1735-1780, 1997
- [4] Saima Aman and Stan Szpakowicz, “Identifying expressions of emotion in text”, In Proceedings of 10th International Conference on Text, Speech and Dialogue, 2007
- [5] M. Purver and S. Battersby, “Experimenting with distant supervision for emotion classification”, In Proceedings of EACL 2012, pp. 482-491, 2012.
- [6] T. Hasegawa, N. Kaji, N. Yoshinaga, M. Toyoda, “Predicting and eliciting addressee’s emotion in online dialogue”, In Proceedings of ACL, pp. 964-972, 2013.

- [7] S. Kang, H. Park, J. Seo, “Emotion classification of user’ s utterance for a dialogue system” , Korean Journal of Cognitive Science, Vol. 21, No. 4, pp. 459-480, 2010.
- [8] 신동원 외, “임베딩 자질을 이용한 대화의 감정 분류” , 제27회 한글 및 한국어 정보처리 학술대회, pp. 109-114, 2015.
- [9] Y. Kim, “Convolutional Neural Networks for Sentence Classification” , Conference on Empirical Methods in Natural Language Processing, 2014.
- [10] J. Lee and Franck Dernoncourt, “Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks” , Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016
- [11] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781, 2013
- [12] X. Glorot and Y. Bengio., “Understanding the difficulty of training deep feedforward neural networks” , In International Conference on Artificial Intelligence and Statistics, 2010.