WWW.JBDTP.ORG

ISSN: 2692-797

JBDTP Professional Vol. 1, No. 1, 2022 DOI: 10.54116/jbdtp.v1i1.19

Four-Class Emotion Classification Problem Using Deep Learning Classifiers

Miaojie Zhou
School of Graduate Professional Studies, Penn State University,
Malvern, PA
mjz5304@psu.edu

Satish Mahadevan Srinivasan
School of Graduate Professional Studies, Penn State University,
Malvern, PA
sus64@psu.edu

Abhishek Tripathi
Department of Accounting and IS,
The College of New Jersey,
Ewing, NJ
tripatha@tcnj.edu

Abstract

Social media sites and blogs generate a vast amount of emotionally rich data in the form of tweets, status updates, blog posts, etc. Such textual data represent emotions expressed by an individual or a group of people on any given topic. By analyzing the emotions within these textual data, we can get an idea about how individuals or communities express their views. Analytical techniques are widely used for analyzing emotions within these texts. However, due to the training datasets' imbalanced nature, the supervised classifiers fail to classify the different emotion classes. As a result, these classifiers demonstrate a poor performance in identifying emotions within the texts. Here, using a constructed heterogeneous training dataset from well-known training datasets we have trained two deep learning models namely the Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) to address a four-class emotion (Anger, Sadness, Happy, and Surprise) classification problem. By appropriately tuning the deep learning classifiers' hyperparameters, our study reveals that the CNN classifier has slightly better performance (77%) than the RNN classifier (76%) for a four-class emotion classification problem.

Keywords *Emotion classification, Deep learning, Supervised classifiers, 10-Fold validation, Word embeddings*

1. Introduction

Human beings have mostly expressed their emotions either by speech or through written texts. With the emergence of social media and blogging sites, individuals and communities have found a way to freely express their opinions, feelings, and thoughts on various topics through texts. Irrespective of the number of characters, texts often hold a wealth of information on how individuals or communities communicate their thoughts, emotions (happiness, anxiety, and depression), and feelings within their network. By analyzing the corpus of texts from the social media and blogging site, one can learn not only the emotions of individuals but also the emotions of larger groups (such as a certain country, state etc.). More commonly expressed, emotions include *anger*, *disgust*, *fear*, *happiness*, *surprise*, *sadness*, *tensed*, etc. For example, the text "I felt quite *happy* and lighthearted; I put on the shoes and danced and jumped about in them" expresses a happy emotion. Not only one, but more than one emotion can be expressed within a text. Since texts lack structure and size, determination of emotions, i.e., *Emotion classification*, is a very challenging task.

Even though sentiment analysis has been widely studied in the field of Data Mining and Machine Learning, it does not address the wide range of emotions that are associated with human behavior. Moreover, it is important to know the exact emotion behind a topic rather than a generic sentiment. Since several different emotions are expressed within a text (sentence), it becomes necessary to analyze each sentence within a document to determine the overall emotion. Ghazi, Inkpen, and Szpakowicz (2010) found that emotion expressions tend to be the most informative in an expressive sentence, so emotion classification is practically important to text summarization. Nowadays, the popularity of short messages within social media and blogging sites are replacing the traditional electronic document. Mining emotions within these short messages is another challenge for emotion classification. Our study is significant for researching the use of deep learning in the short, textual data and the applicability and practical use across domains in actual life, such as the social network posts and movie review analysis. Coviello *et al.* (2014) found that online social networks may magnify the intensity of global emotional synchrony. Determining the emotional category on IMDB datasets accurately predicts the human's preference/interest in movies (Liu 2020), which would further affect film studio and cinema's decision-making in next quarter (Liu 2020).

Unlike conventional texts, short messages are peculiar in structure and size. Adding to that is the language used by people within these texts to express their emotions which is very different from the digitized documents (Ling and Baron 2007). A major challenge is also posed by the availability of many features within the texts. There are also challenges associated with manually classifying the texts into different emotion types. Manually annotating the texts may be ambiguous at times and does not guarantee complete accuracy (Hasan, Rundensteiner, and Agu 2014). Also, the inherent nature of the different types of emotions makes it very difficult to differentiate between them. According to the Circumplex model, there are 28 affect words or emotions. In this model, several emotions are clustered so close to each other that it becomes very hard to differentiate between them. There is always a high probability of mislabeling the emotions that are clustered so close to each other. For example, sadness and depression are two emotions that are very close to each other, that it is hard to differentiate between them (Russell 1980). All these factors together inhibit the classifiers from learning the critical features that can enable it to identify emotions within the text.

Previous studies have focused on exploring the potentiality of different classification techniques using the bag-of-words (BOW) and/or the *n*-grams as features (Aman and Szpakowicz 2007; Diman, Inkpen, and Szpakowicz 2010; Chaffar and Inkpen 2011; Hasan, Rundensteiner, and Agu 2014; Badshah *et al.* 2016). These studies have used traditional machine learning classifiers, but none have explored the deep learning classifiers' potentiality for emotion classification.

In this paper, we have demonstrated both the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) classifiers' potentiality for the four-class emotion classification problem. We have achieved an average of 76–77% accuracy when classifying 277 instances in the testing dataset. According to the observational results, CNN slightly outperformed RNN in classifying all the four emotion types. Still, they both misclassified a significant number of instances from the surprise class to the happy class.

The rest of the paper is organized as follows. In section 2, we report the survey of the literature. In section 3, we detail the materials and methods employed in this study. In section 4, we present the results from this study and discuss our findings. Finally, in section 5, we conclude the paper and outline the future direction of our research.

2. Related Work

While classifying emotions based on textual data is a relatively new research area, it has attracted lots of attention. Bhowmick, Basu, and Mitra (2010) observed the same relative performance exhibited by humans and machines on different data sets in the emotion classification task, so the high accuracy achieved by machine learning or deep learning could be trusted. Nowadays, approaches employing Deep neural networks have been widely studied for emotion classification in textual data. This technological advancement significantly outperforms other off-the-shelf models (Chatterjee *et al.* 2019). CNNs have been a popular choice in several different works. Simple CNN with slight hyperparameter tuning demonstrated excellent results on multiple benchmarks, including the fine-grained Stanford Sentiment Treebank (SST) for binary classification (Kim 2014). Kalchbrenner, Grefenstette, and Blunsom (2014) have discussed using a Dynamic CNN (DCNN) for Twitter sentiment prediction. According to them, the

DCNN can handle varying lengths of input sentences and is easily applicable to any language due to the usage of Dynamic k-Max Pooling, which reduced the error rate by 25% (Kalchbrenner, Grefenstette, and Blunsom 2014). Acharya *et al.* (2018) have proposed a complex 13-layer CNN architecture for emotion detection in EEG signals.

On the other hand, RNNs are designed to handle sequence problems and have gained much attention over time. Lai *et al.* (2015) introduced a recurrent CNN that automatically judges and captures the key components in texts that can help boost the experiments' accuracy. Abdul-Mageed and Ungar (2017) have proposed a core model of Gated RNNs (GRNNs) and a modern variation of RNN for classifying emotions in several dimensions with high accuracies. Kratzwald *et al.* (2018) have reported that both RNN and sent2affect (transfer learning) consistently outperform the traditional machine learning algorithms across six benchmark datasets.

Wang *et al.* (2016) have proposed a regional CNN-LSTM model to predict the VA ratings of texts in the SST corpus. Zhang *et al.* (2019) have proposed a Coordinated CNN- LSTM-Attention (CCLA) model using the SoftMax regression classifier on four datasets including Movie Review Data (MR), Large movie review (IMDB), TREC question dataset (TREC), and Subjectivity dataset (SUBJ). Socher *et al.* (2013) have introduced Recursive Neural Tensor Network (RNTN) for the famous SST dataset. They have reported 85.4% accuracy for sentiment classification. Irsoy and Cardie (2014) have proposed a deep recursive neural network constructed by stacking multiple recursive layers very similar to the conventional deep feed-forward networks. As a result, they achieved 50% accuracy for the task of fine-grained sentiment classification (five-classes of SST dataset; Irsoy and Cardie 2014).

Several other studies have been reported relating to textual-based emotion classification. Jabreel and Moreno (2019) have proposed a novel method, Binary Neural Network (BNet) for emotion classification on Twitter data (SemEval2018 Task 1: E-c multi-label emotion classification) and have reported an accuracy score of 59%. Zheng *et al.* (2014) have trained a deep belief network (DBN) and have achieved an accuracy of 86.91% and 87.62% in the experiments of DBN and DBN-HMM models, respectively. Zhou *et al.* (2016) propose a BLSTM architecture that helped capture long-term sentence dependencies and introduced a combined model BLSTM-2DCNN, which achieves 52.4% accuracy on SST binary classification and fine-grained classification tasks. Hamdi, Rady, and Aref (2020) utilized CNN streams and the pretrained word embeddings (Word2Vec) and achieved 84.9% accuracy from the Stanford Twitter Sentiment dataset. Zhang, Lee, and Radev (2016) presented the Dependency Sensitive CNNs (DSCNNs) that outperforms traditional CNNs and achieved 81.5% accuracy in the sentiment analysis of Movie Review Data (MR) proposed by Pang and Lee (2005). Zhou *et al.* (2015) proposed a novel model called C-LSTM for sentence representation utilizing both CNN and LSTM to achieve 49.2% in five-class classification tasks.

In the next section, we discuss the materials and methods employed in this study.

3. Materials and Methods

3.1 Dataset

Here we use a synthetic dataset SYN80 constructed by combining the Alm's dataset (Alm 2008; Chaffar and Inkpen 2011) and Aman's dataset (Aman and Szpakowicz 2007; Chaffar and Inkpen 2011). Instances from both the Alm's and Aman's datasets were combined and reshuffled. After reshuffling, 80% of the instances were randomly picked without replacement to create the synthetic dataset SYN80. The SYN80 dataset contains instances, each represented as *{text, emotion-class}*. We ensured that all the instances in this dataset have a single annotation of an emotion class (Srinivasan and Ramesh 2018). In SYN80, both with resampling and without resampling, the performance of the traditional classifiers was significantly boosted with few exceptions for a four-class emotion classification problem (Srinivasan and Ramesh 2018). Table 1 lists the class-wise distribution of the number of instances in the SYN80 dataset.

3.2 Models

The deep neural networks have the ability to learn and model nonlinear and complex relationships, which is important because, in the context of the emotion classification problem, many of the relationships between inputs and outputs are nonlinear as well as complex (Kalchbrenner, Grefenstette, and Blunsom 2014; Abdul-Mageed and Ungar 2017; Chatterjee *et al.* 2019; Jabreel and Moreno 2019; Zhang *et al.* 2019; Hamdi, Rady, and Aref 2020). Therefore,

Table 1: Class-wise distribution of the number of instances in the SYN80 dataset.

Class	Happy	Anger	Sadness	Surprise
Number of sentences	739	133	325	186

in this study, we have explored both the CNNs and RNNs architecture and have compared the results from these two classifiers. To optimize the hyperparameters, we employed the grid search technique. As discussed in literature (Khorrami *et al.* 2016; Lakomkin, Bothe, and Wermter 2018; Al Machot *et al.* 2019; Ghosal *et al.* 2019), upon performing the grid search, we determined the optimal numbers for the filters and layers for both the CNN and RNN models. The grid search functionality was implemented in Python using the GridSearchCV class in the Scikit-learn library. An exhaustive list of values was provided for tuning the hyperparameters. The performance of the models was compared at different stages to finalize the values for the filters and layers for both CNN and RNN.

3.2.1 CNNs

CNNs is a network that employs convolution operation in at least one layer that works well, especially in image data or a grid of values. Convolution is a specialized kind of mathematical operation that can be used in one-dimension, two-dimension, and multi-dimension. The convolution layer abstracts or scales the input matrix to a feature map using multiplication or dot product. As mentioned in the related work section, CNN is an improved and developed neural network. The CNN architecture employed in this research consists of five different types of layers. (1) The embedding layer that consists of pretrained weights for the words and maps each input word to a 300-dimensional vector. (2) Convolutional layer that consists of filters (kernels) which slide across the embedding layer; this layer is used to obtain the feature map. (3) MaxPooling layer or down-sampling layer that performs the maxpooling operation to reduce the dimension of output neurons and computational intensity, thus preventing overfitting; the maxpooling operation selects only the maximum value in each feature map. (4) Dense (fully connected) layer that has a full connection to all the activations in the previous layer. (5) And flatten layer in Keras that reshapes the tensor to have a shape that is equal to the number of elements contained in the tensor. The architecture of the CNN model and the number of neurons in each layer is shown below (Table 2).

The CNN network consists of 3 convolution layers with MaxPooling after each layer. For the first two convolutional layers, the filters were set to 150, kernel size was specified to 2, padding used was ‘same,’ and the activation function used was ‘ReLU.’ However, for the third convolutional layer the filters were set to 128. The pool size was set to 5x5 for the first two MaxPooling layers. The third MaxPooling layer does global pooling with a size of 24. Then we added a flatten layer whose output was then fed to a Dense layer consisting of 50 neurons. Since this CNN is classifying four categories, the output layer was set to four outputs. For the loss function, we used the cross-entropy, and Adam optimizer was set to minimize the loss.

Here we use two activation functions: (1) ReLU and (2) SoftMax. As highly recommended, we have used an activation function after every convolutional layer. The leaky rectifier linear unit (LeakyReLU) was used as an activation function for the convolutional layers 2, 4, and 6 to add nonlinearity and sparsity in the network structure. We also used the SoftMax activation function in the output layer to boost the performance of the multi-class classification.

3.2.2 RNNs

RNNs are networks that process a sequence of values and have a chain-like structure which means that connections between nodes form a directed graph along a temporal sequence. RNNs have the capability to scale to much longer sequences. According to the literature review, RNN is widely used in emotion classification problems due to its temporal dynamic property. It uses internal memory to process the sequence of inputs and can process sequences of variable length.

Table 2: The details of the CNN structure used in this research.

Layers	Type	Number of Neurons (Output Shape)
1	Embedding	(600,300)
2	Convolution	(600,150)
3	MaxPooling	(120,150)
4	Convolution	(120,150)
5	MaxPooling	(24,150)
6	Convolution	(24,128)
7	MaxPooling	(1,128)
8	Flatten	128
9	Dense (fully connected)	50
10	Dense (fully connected)	4

Table 3: The details of the RNN structure used in this research.

Layers	Type	Number of Neurons (Output Shape)
1	Embedding	(600,300)
2	LSTM	128
3	Dense (Fully Connected)	64
4	Dropout	64
5	Dense (Fully Connected)	4

One of the most important layers in RNN is the LSTM layer, which has shown success in various other domains. We use one LSTM layer after the embedding layer to processes the text from left to right. Deep neural networks tend to run the risk of overfitting, especially when the training dataset is small. Consequently, we use the dropout parameter within the recurrent layer, which is equivalent to randomly dropping out connections between the recurrent LSTM cells. We also use a dropout layer between the output of a 64-layer dense network and the output layer. The architecture of the RNN model and the number of neurons in each layer are shown below (Table 3).

A fiev-layer RNN has been employed in this study. The dimension of word embeddings used here is 300, the number of hidden units in the LSTM layer is 128. Within the LSTM layer, we set the dropout and the recurrent dropout parameter to 0.2. We have used ReLU as the activation function in the first dense layer with 64 hidden units. For regularization, we employ the Dropout operation with a dropout rate of 0.5. The final output layer (dense layer) consists of four outputs since we address a four-class classification problem. We also use SoftMax as the output layer. The loss function used here is the cross-entropy and Adam optimizer, which is set to minimize the loss.

In the next section, we discuss the experimental procedures performed in this study.

3.3 Experimental Designs

3.3.1 Preprocessing

The preprocessing begins with shuffling the instances in the SYN80 dataset to maintain randomness in classes before the model is trained. The textual instances were corrected for spelling mistakes. Then, all the stop words and special characters were removed. All the textual instances were converted into lower case. The instances were then padded to a length of 600 characters. Instances that were longer than 600 characters were trimmed. Next, we performed tokenization and retained the top 1500 words for our analysis. We used the Global Vectors for Word Representation (GloVe; Pennington, Socher, and Manning 2014) to initialize the weights for these words and then created the embedding matrix.

3.3.2 Ten-fold cross-validation

A 10-fold cross-validation approach was followed in this study. First, the preprocessed dataset was split into the train and test set following the ratio 80:20, respectively. When training the model, we performed 10-fold cross-validation using the training dataset. For performing the 10-fold cross-validation technique, we initially partitioned the training dataset into 10-folds. Across the 10 different experiments performed, in each experiment, 9-folds of the dataset were used for training the model, and the remaining 1-fold was used for validating the model. Finally, the model was tested on the test dataset. In the testing phase, we performed 10 different experiments using 10 different models and then averaged the resulting metric accuracy. For each experiment, we used the same test dataset but across different models. Here we report the validation and test accuracy, sensitivity(recall), F1-score, and precision all averaged across 10 different experiments.

3.3.3 Performance measures

In this study, we have tried to address a four-class classification problem. Here, we report the values for *Accuracy*, *Recall*, *Precision*, and *F1-score*. *Accuracy* is the ratio of the number of truly predicted samples to the total number of samples in the test dataset. *Precision* is the fraction of relevant instances among the retrieved instances and *Recall* is the fraction of the relevant instances that have been retrieved over the total number of relevant instances. *F1-score* is the harmonic mean of precision and recall. All the above discussed performance measures were computed using the confusion matrix.

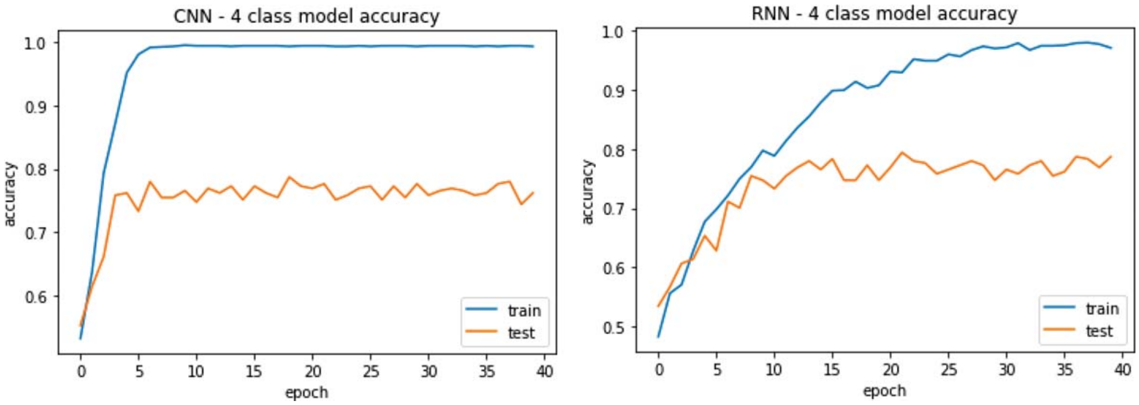

Figure 1: CNN (left) and RNN (right) learning curve for four-class emotion classification.

Table 4: Averaged performance measure (accuracy) for both the CNN and ANN on SYN80 dataset.

	Model	Validation Accuracy	Test Accuracy
(1)	CNN	75.68%	77%
(2)	RNN	73.79%	76%

Both the CNN and RNN networks were created using the *Keras* package in Python. To summarize, this study conducts a series of experiments based on the steps outlined below:

1. All the textual instances were cleaned/preprocessed as outlined in section 3.3.1. Using the shuffle functionality, we randomized the instances within the SYN80 dataset. The resultant dataset was then split into the ratio of 80:20; 10-fold cross-validation was performed on the 80% (training) dataset.
2. The training dataset instances were transformed into a feature vector and the embedding matrix was constructed using the steps outlined in section 3.3.1.
3. Both the CNN and RNN network was trained using the training dataset, and their performance across 10-fold cross-validation was recorded as outlined in section 3.3.2.
4. Finally, we test our models using the test dataset and report the performance measures (see section 3.3.3) across the validation and test dataset averaged over 10 different experiments as discussed in section 3.3.2.

In the next section, we present the results and discussions from this study.

4. Result and Discussion

A four-class emotion classification was performed using both the CNN and RNN. we noticed that the CNN model's test accuracy tends to be stable after 10 iterations (see Figure 1, left). Therefore, we choose to set 10 epochs for the CNN model. The RNN model established the test accuracy after 15 iterations (see Figure 1, right).

The overall average prediction accuracy on validation and test dataset for both the CNN and RNN are reported in Table 4. CNN reported an average of 75.68% validation accuracy slightly better than the RNN classifier (see Table 4). The CNN reported an average of 77% accuracy for the four-class emotion classification in the test dataset. Based on the validation and test accuracy for both the CNN and RNN, we can conclude that there is no evidence of overfitting. It is also clearly evident that both the classifiers performed significantly better in the test dataset than in the validation dataset (see Table 4). A total of 100 iterations of training was performed to train the CNN models. However, it took 150 iterations to train the RNN models. It is also important to note that the time taken to train the RNN models was significantly larger than the time taken to train the CNN models (see Figure 1).

Srinivasan and Ramesh (2018) have reported a baseline performance of the supervised classifiers including k-nearest neighbor (kNN; k = 1, 3, 5, 7), J48 Classifier (C4.5), Classification and Regression Trees (CART), Naïve Bayes Multinomial (NBM), Random Forest (RF), and Sequential Minimal Optimization (SMO) on the SYN80 dataset for a four-class emotion classification problem. The test accuracies on the SYN80 dataset without

Table 5: Averaged performance measure (precision, recall and F1-score) for CNN and RNN on the validation dataset.

	Model	Class	Class Name	Precision	Recall	F1-Score
(1)	CNN	0	Anger	0.806	0.694	0.739
		1	Sadness	0.661	0.776	0.712
		2	Happy	0.849	0.856	0.85
		3	Surprise	0.654	0.528	0.578
(2)	RNN	0	Anger	0.717	0.7	0.704
		1	Sadness	0.679	0.748	0.703
		2	Happy	0.832	0.87	0.851
		3	Surprise	0.652	0.432	0.51

Table 6: Averaged performance measure for CNN and RNN on the test dataset.

				Predicted				
	Model	Class Name	Actual	0	1	2	3	Class Accuracy
(1)	CNN	Anger	0	22.8	4.1	2.8	3.3	0.9426
		Sadness	1	2	42.7	7.4	2.9	0.8747
		Happy	2	1.1	13.5	125.9	6.5	0.8422
		Surprise	3	2.6	4.8	12.4	22.2	0.8827
(2)	RNN	Anger	0	23	3.5	5.3	1.2	0.9303
		Sadness	1	3.2	41.1	7.4	3.3	0.8755
		Happy	2	0.9	12	128	6.1	0.8368
		Surprise	3	5.2	5.1	13.5	18.2	0.8758

resampling were 61.61%, 55.39%, 53.51%, 53.58%, 55.24%, 68.33%, 74.48%, 73.61%, and 76.72% for the classifiers 1NN, 3NN, 5NN, 7NN, C4.5, CART, NBM, RF, and SMO, respectively (Socher *et al.* 2013). The test accuracies for CNN (77%) and RNN (76%) obtained in this study are comparable to the performance of the SMO classifier. When compared against traditional classifiers such as kNN, C4.5, CART, NBM, and RF the performance of the CNN and RNN are slightly superior. Here it is important to note that the preprocessing step did not involve resampling the SYN80 dataset.

In the validation dataset, both the CNN and RNN demonstrated a high F1-score for the *happy* emotion. Both CNN and RNN demonstrated a significantly better average F1-score (above 70%) for the *anger* emotion with CNN significantly outperforming the RNN classifier. However, both the classifier recorded a significantly low F1-score for the surprise emotion compared to the other emotion classes with CNN significantly outperforming the RNN (see Table 5). This is a very striking observation as both the emotion classes, i.e., *anger* and *surprise*, had significantly smaller number of instances in the *SYN80* dataset (see Table 1). To best reason this observation, we believe that the instances belonging to the *anger* class have a rich set of features that can discriminate these instances with respect to the other classes. For the *surprise* class, the RRN classifier resulted in a higher number of false negatives than CNN, which suggests that the features within the instances belonging to the *surprise* class are not clearly discriminative (see Table 5).

The average class accuracy across each emotion class over 10 different models using a single test dataset is reported in Table 6. We also report the averaged confusion matrix of CNN and RNN on the single test dataset over 10 different models. In average 32% of the instances (13.5 out of 42) belonging to the *surprise* emotion has been misclassified as *happy* by the RNN classifier (see Table 6). On the other hand, CNN also misclassified 29.5% of the *surprise* instances (12.4 out of 42) on an average to *happy*. This observation on the test dataset is very consistent with the observations, i.e., low recall and low precision, in the validation dataset (see Tables 5 and 6).

5. Conclusion

In this study, we have demonstrated the deep neural networks' potentiality for the four-class emotion classification problem. Both the CNN and RNN classifiers were explored in this study. On the SYN80 dataset, we achieved an average of 77% accuracy while trying to classify 277 instances belonging to four different emotion types. CNN slightly outperformed RNN in classifying all the four emotion types. However, both the classifiers misclassified a significant number of instances from the surprise class to the happy class. This observation can be attributed to the fact that both the emotion types, i.e., surprise and happy, are very similar to each other, which is also evident from the circumflex model, and the instances belonging to the surprise type clearly lacks distinctive features that can help the models to discriminate this emotion against the happy emotion. In this study, we have limited our investigations to only four classes of emotions. In the future, we plan to consider additional emotion types and investigate the potentiality of the deep neural networks for emotion classification.

References

Abdul-Mageed, Muhammad, and Lyle Ungar. 2017. "Emonet: Fine-Grained Emotion Detection with Gated Recurrent Neural Networks." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long papers)*, 718–28.

Acharya, U., Shu Lih Oh Rajendra, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. 2018. "Deep Convolutional Neural Network for the Automated Detection and Diagnosis of Seizure Using EEG Signals." *Computers in Biology and Medicine* **100**: 270–78. doi: 10.1016/j.compbiomed.2017.09.017

Alm, Ebba Cecilia Ovesdotter. 2008. *Affect in Text and Speech*. Urbana*:* University of Illinois at Urbana-Champaign, ProQuest Dissertations Publishing, 1–119.

Al Machot, F., A. Elmachot, M. Ali, E. Al Machot, and K. Kyamakya. 2019. "A Deep-Learning Model for Subject-Independent Human Emotion Recognition Using Electrodermal Activity Sensors." *Sensors* **19**, no. 7: 1659, 1–14. doi: 10.3390/s19071659

Aman, Saima, and Stan Szpakowicz. 2007. "Identifying Expressions of Emotion in Text." In: Matoušek V., Mautner P. (eds) *Text, Speech and Dialogue. TSD 2007. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg 4629: 196–205.

Badshah, Abdul Malik, Jamil Ahmad, Mi Young Lee, and Sung Wook Baik. (2016) "Divide-and-conquer based ensemble to spot emotions in speech using MFCC and random forest." In *the proceedings of the 2nd international integrated conference & concert on convergence*, 1-8. *arXiv*:1610.01382.

Bhowmick, Plaban Kumar, Anupam Basu, and Pabitra Mitra. 2010. "Classifying Emotion in News Sentences: When Machine Classification Meets Human Classification." *International Journal on Computer Science and Engineering* **2**, no. 1: 98–108.

Chaffar, Soumaya, and Diana Inkpen. 2011. "Using a heterogeneous dataset for emotion analysis in text." in *Canadian conference on artificial intelligence*, LNAI 6657, 62–67.

Chatterjee, Ankush, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019. "Understanding Emotions in Text Using Deep Learning and Big Data." *Computers in Human Behavior* **93**: 309–17. doi: 10.1016/j.chb.2018.12.029

Coviello, Lorenzo, Yunkyu Sohn, Adam DI Kramer, Cameron Marlow, Massimo Franceschetti, Nicholas A. Christakis, and James H. Fowler. 2014. "Detecting Emotional Contagion in Massive Social Networks." *PLoS One* **9**, no. 3: e90315, 1–6. doi: 10.1371/journal.pone.0090315

Diman, Ghazi, Diana Inkpen, and Stan Szpakowicz. 2010. "Hierarchical versus Flat Classification of Emotions in Text." *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 140–6.

Ghazi, Diman, Diana Inkpen, and Stan Szpakowicz. 2010. "Hierarchical Versus Flat Classification of Emotions in Text." *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, **1**(1), 140–6.

Ghosal, D., N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh. 2019. "Dialoguegcn: A Graph Convolutional Neural Network for Emotion Recognition in Conversation." *arXiv:1908.11540*. doi: 10.48550/ARXIV.1908.11540, 1–11.

Hamdi, Eman, Sherine Rady, and Mostafa Aref. 2020. "A Deep Learning Architecture with Word Embeddings to Classify Sentiment in Twitter." *International Conference on Advanced Intelligent Systems and Informatics*, 115–25.

Hasan, Maryam, Elke Rundensteiner, and Emmanuel Agu. 2014. "Emotex: Detecting Emotions in Twitter Messages." *ASE Bigdata/SocialCom/Cybersecurity Conference*, Stanford University, 1–10.

Irsoy, Ozan, and Claire Cardie. 2014. "Deep Recursive Neural Networks for Compositionality in Language." *Advances in Neural Information Processing Systems* **27**: 2096–104.

Jabreel, Mohammed, and Antonio Moreno. 2019. "A Deep Learning-Based Approach for Multi-Label Emotion Classification in Tweets." *Applied Sciences* **9**, no. 6: 1123, 1–16. doi: 10.3390/app9061123

Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. 2014. "A Convolutional Neural Network for Modelling Sentences." *arXiv*:1404.2188v1. doi: 10.48550/ARXIV.1404.2188, 1–11.

Khorrami, P., T. Le Paine, K. Brady, C. Dagli, and T. S. Huang. 2016. "How Deep Neural Networks Can Improve Emotion Recognition on Video Data." *2016 IEEE International Conference on Image Processing (ICIP)*, 619–23. IEEE.

Kim, Yoon. 2014. "Convolutional Neural Networks for Sentence Classification." *arXiv*:1408.5882v2. doi: 10.48550/ARXIV.1408.5882, 1–6

Kratzwald, Bernhard, Suzana Ilić, Mathias Kraus, Stefan Feuerriegel, and Helmut Prendinger. 2018. "Deep Learning for Affective Computing: Text-Based Emotion Recognition in Decision Support." *Decision Support Systems* **115**: 24–35. doi: 10.1016/j.dss.2018.09.002

Lai, Siwei, Liheng Xu, Kang Liu, and Jun Zhao. 2015. "Recurrent Convolutional Neural Networks for Text Classification." *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, **1**(1), 2267–2273.

Lakomkin, E., C. Bothe, and S. Wermter. 2018. "GradAscent at EmoInt-2017: Character-and Word-Level Recurrent Neural Network Models for Tweet Emotion Intensity Detection." *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, *arXiv:1803.11509*. doi: 10.18653/v1/W17-5222"10.18653/v1/W17-5222, 169–74.

Ling, Rich, and Naomi S. Baron. 2007. "Text Messaging and IM: Linguistic Comparison of American College Data." *Journal of Language and Social Psychology* **26**, no. 3: 291–98. doi: 10.1177/0261927X06303480

Liu, Bing. 2020. "Text Sentiment Analysis Based on CBOW Model and Deep Learning in Big Data Environment." *Journal of Ambient Intelligence and Humanized Computing* **11**, no. 2: 451–8. doi: 10.1007/s12652-018-1095-6

Pang, Bo, and Lillian Lee. 2005. "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales." *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05) arXiv:cs/0506075*. doi: 10.3115/1219840.1219855, 115–24.

Pennington, Jeffrey, Richard Socher, and ChristopherD. Manning. 2014. "Glove: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–43.

Russell, James A. 1980. "A Circumplex Model of Affect." *Journal of Personality and Social Psychology* **39**, no. 6: 1161–78. doi: 10.1037/h0077714

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. "Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank." *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–42.

Srinivasan, Satish M., and Prashanth Ramesh. 2018. "Comparing Different Classifiers and Feature Selection Techniques for Emotion Classification." *International Journal of Society Systems Science* **10**, no. 4: 259–84. doi: 10.1504/IJSSS.2018.095595

Wang, Jin, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016. "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 225–30.

Zhang, Rui, Honglak Lee, and Dragomir Radev. 2016. "Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents." *Proceedings of the 2016 Conference of the North America Chapter of the Association for Computational Linguistics: Human Language Technologies. arXiv:1611.02361*. doi: 10.18653/v1/N16-1177"10.18653/v1/N16-1177, 1512–21.

Zhang, Yangsen, Jia Zheng, Yuru Jiang, Gaijuan Huang, and Ruoyu Chen. 2019. "A Text Sentiment Classification Modeling Method Based on Coordinated CNN-LSTM-Attention Model." *Chinese Journal of Electronics* **28**, no. 1: 120–26. doi: 10.1049/cje.2018.11.004

Zheng, Wei-Long, Jia-Yi Zhu, Yong Peng, and Bao-Liang Lu. 2014. "EEG-Based Emotion Classification Using Deep Belief Networks." *2014 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.

Zhou, Peng, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. "Text Classification Improved by Integrating Bidirectional LSTM with Two-Dimensional Max Pooling." *arXiv:1611.06639v1*. doi: 10.48550/ARXIV.1611.06639, 1–11.

Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. "A C-LSTM Neural Network for Text Classification." *arXiv:1511.08630*. doi: 10.48550/ARXIV.1511.08630, 1–10.