Sentiment and Emotion Classification using Convolutional Neural Networks with Long Short Term Memory (CNN-LSTM)

Dr. M. Subba Rao, Professor & Head**,** Department of Computer Science and Engineering, Annamacharya Instituteof Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India-516126.

S. Deepa, A. Charitha, D. Manoj kumar, N. Sai Venkat J.NagaSree, Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India-516126

ABSTRACT:

As a result of increase in internet usage, there is a massive amount of information available to web users, as well as a massive amount of new information being created daily. To facilitate internet pick-up, trading ideas, and disseminating assessments, the internet has evolved into a stage of large volumes of data. Facebook, Online product Reviews and Twitter generate a lot of data every day. As a result, text handling is crucial in making decisions. Sentiment analysis has surfaced as a method for analyzing Twitter data. In this paper, we collected a Kaggle dataset on multimedia-text information. It contains three variants: neutral, positive, negative. First, we used Deep Learning methods to clean the text data. Later, we applied Convolutional neural networks and recurrent neural networks with LSTM techniques for classifying multimedia-text in three different ways: positive, negative sentiment analysis. As we didn’t want the neutral so we dropped the neutral and only considered the positive and negative sentiment. We achieved a good accuracy for the classification of positive and negative.

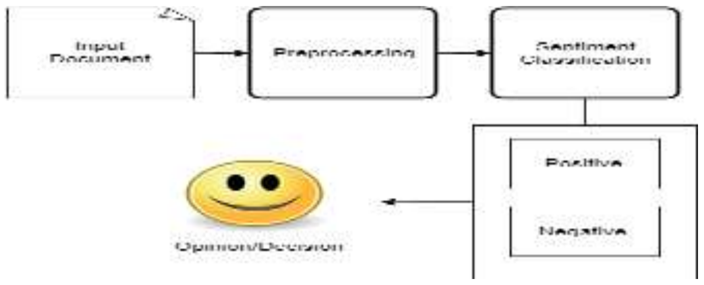
INDEX TERMS:

Deep Learning, Sentimental Analysis, Convolutional Neural Network, Long-Short Term Memory, Recurrent neural networks.

1. INTRODUCTION:

1.1. MOTIVATION

The popularization of the Internet has brought the extreme convenience of information exchange. Hot issues can trigger a great quantity discussion on the Internet in a short time. The collection, analysis and response of the public opinion is called public opinion analysis. Its key technology is emotion classification, which is an important subtask in natural language processing. Because of the top trending search mechanism and public discussion feature of social media, the objects of online public opinion analysis are selected from Twitter, Weibo and other social media. Text emotion classification technology has developed by leaps and bounds over the years. Since bidirectional encoder representation from transformers (BERT) [1] published in 2018, the state of the art of natural language processing (NLP) each sub-tasks are monopolized by models based on Transformer and pre-training. That is because, on the one hand, the Transformer network has good parallel computing ability and has the advantages such as coding by location are very suitable for NLP tasks [2], on the other hand, the two- stage mode of sub-tasks after the pre-training can improve the effect of text processing.



It is difficult for CNN series network to capture long distance features, and the pooling layer cannot capture word location information. RNN series network is difficult to be used in large-scale parallel computing. Transformer networks solves the various issues of CNN and RNN [3] to achieve such excellent results for various NLP sub-tasks. However, this does not mean that Transformer network is perfect. Taking XLNet [4] as example, the premise of achieving extremely high accuracy is the support of big data, which means the demand for data scale and hardware. Although many scholars have made lightweight improvement work, training related networks still requires high hardware configuration and plenty of time. Hence, RNN and CNN series networks are still the choices with higher priority in lightweight requirements [5]–[9].For such short-term and small-scale sentiment classification tasks, the classification model is generally selected as a deep neural network with fewer layers, including TextCNN[5], CharCNN [6], FastText [7], TextRNN [8], TextRCNN [9], etc. Although these networks do not require numerous training time and large-scale corpus data, their classification effect is not good even after optimization and tuning, due to the limitation of the model structure.

In order to improve the effect of sentiment classification under small data scale, a variety of methods have been applied to expand the information contained in the data. One of the simplest ideas is to expand the data size. Sun and He [10] proposed a hybrid neural network model based on data expansion technology. The data expansion technology can improve the data scale, enhance the generalization performance of the model, and then improve the accuracy. However, the technique only artificially expands the data scale, while long- term training on large-scale data is still required during model training.Another feasible and popular method is to use multi-modal information. Kumar et al. [11] used a combination of pictures and text in social media to train a hybrid classifier and achieved higher accuracy than using two types of information alone. Bairavel and Krishnamurthy [12] added video information on this basis, and extracted features from the information of the three modalities to train a neural network with very high accuracy.

1. The user attributes in social media are added to the sentiment classification model, and thus the input information is easy to obtain.
2. Because the CNN network used to extract user attributes features has fewer parameters, the model training time is short.
3. Multimodal social media text information sentiment classification algorithm is the state-of-the-art in short- term and small-scale data sentiment classification.

Mairesse et al. [13] divided users into five personalities based on attribute characteristics, and added different weights for each personality in different emotions to assist text emotion classification, proving that user attributes can be used to assist text emotion classification. In addition to the emotional portrait of users, there are also studies on emotional similarity in time and space. Mitchell et al. [14] combined the geo-tagged text set of social media with the corresponding emotional characteristics of the annual survey of more than 400 cities in the United States, proving that people in a region have certain similarities in word usage and emotional tendency. Li et al. [15] studied people’s collective response to special events in a very short time through social media, and the emotional trend of the public towards an event in a short time is always similar. This project deals with emotion and sentiment analysis in computational literary studies.

2.DEEP LEARNING MODELS:

2.1. Existing System

2.1.1 Deep Learning in Sentiment and Emotion Classification:

Deep learning has recently gained a lot of interest and popularity in many fields of research. In present project, a well-known deep learning network called, the Convolution Neural Network (CNN) is used to sentiment and emotion classification to exploit the benefits of functional learning and classification. A convolution network is a type of neural network that has been investigated since the 1990s and is primarily used to process images, videos, and any type of spatially organized data. In general, we can divide a neural network of convolutions into two well-defined components: the feature extraction part and the classification part.

2.1.2CNN Model for Sentiment and Emotion Classification:

CNN based on are deep learning feature extraction models that have recently been shown to be quite effective in image recognition. As of today, numerous business giants like Google, Twitter, and Amazon are using the templates. And recently, Google researchers applied CNN to video data. It is achieved by performing separate transformations of the filter values as trainable weights in the image. Several filters are added to each path, and they form feature maps along with neural activation functions. A grouping scheme accompanies this, where only the relevant details of the function maps are grouped. The design of the Convolutional Neural network, as seen in Figure 3.1, is composed of input, Convolution layer, clustered layer, fully connected layer, and output layer. The CNN networks have diverse architectures with a specific number of levels of Convolution and grouping. The function map of layer *i* can be represented as the Convolution method: Mi = f(M*i*−1 ⊗ W*i* + bi). W*i* is the weight vector of the transfer core of layer *i*, the process representation describes the transfer phase and is the displacement variable of layer *i*, and (z) is the excitation function. The Convolution nucleus creates new characteristics in the Convolution cycle by performing the Convolution procedure with the characteristics of the inputs.

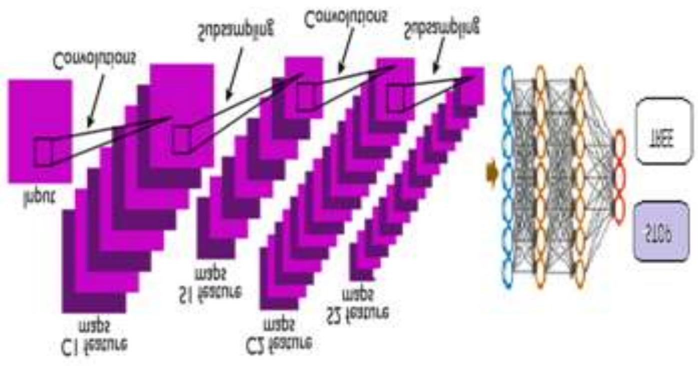


Figure 2.1: CNN model for Texture Classification

2.1.3 Recurrent Neural Network (RNN):

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. The term “recurrent neural network” is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both finite impulse and infinite impulse recurrent networks can have additional stored states, and the storage can be under direct control by the neural network. The storage can also be replaced by another network or graph, if that incorporates time delays or has feedback loops. Such controlled states are referred to as gated state or gated memory, and are part of long short-term memory networks (LSTMs) and gated recurrent units. This is also called Feedback Neural Network (FNN).

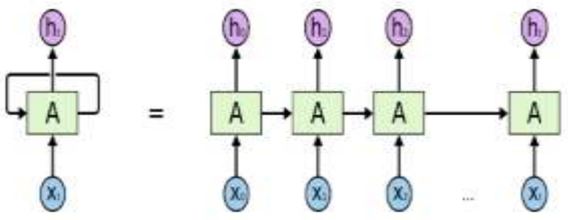


Figure.2.2: An unrolled recurrent neural network

2.1.4. LSTM Networks:

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn! All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single network layer as shown in 3.3.

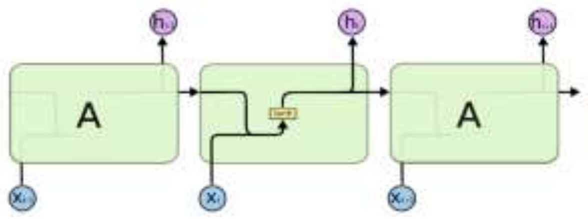


Figure.2.3: The repeating module in a standard RNN contains a single layer

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way in the figure shown in 3.4..

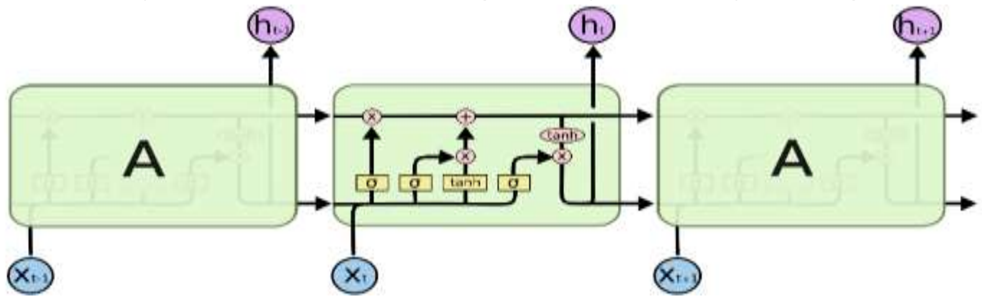


Figure.2.4: The repeating module in an LSTM contains four interacting layers

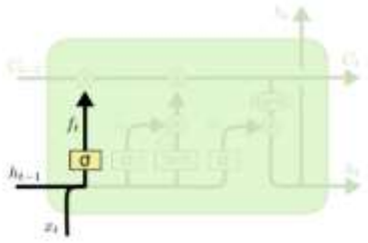
In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent point wise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

2.1.5. The Core Idea behind LSTMs:

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. Sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!" LSTM has three of these gates, to protect and control the cell state.

2.1.6Step-by-Step LSTM Walk Through:

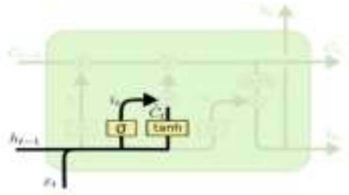
The first step in our LSTM in figure 2.5 is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at ht−1 and xt, and outputs a number between 0 and 1 for each number in the cell state Ct−1. A 1 represents "completely keep this" while a 0 represents "completely get rid of this."



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure.2.5: Step 1 of LSTM

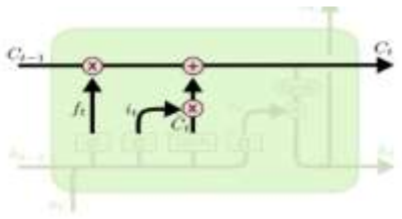
The next step as shown in figure 2.6. is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tan h layer creates a vector of new candidate values, C~t, that could be added to the state. In the next step, we'll combine these two to create an update to the state. In the example of our language model, we'd want to add the gender of the new subject to the cell state, to replace the old one we're forge



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure.2.6: Step 2 of LSTM

It's now time to update the old cell state, Ct−1, into the new cell state CT. The previous steps already decided what to do, we just need to actually do it. We multiply the old state by ft, forgetting the things we decided to forget earlier. Then we add it∗C~t. This is the new candidate values, scaled by how much we decided to update each state value as shown in figure 2.7. In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information in the previous steps.



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

Figure 2.7:Step3 of LSTM

Before diving into the Natural language processing, let us first revisit some concepts of Neural Network. In a regular Neural Network, there are three types of layers:

Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to total number of features in our data (number of pixels in case of an image).
Hidden Layer: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.
Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or Softmax which converts the output of each class into probability score of each class.

2.2. Proposed System:

The CNN - LSTM fusion exploits the invariant features learning of CNN and high accuracy of separation of feature vectors with CNN and SVM.

2.2.1 CNN-LSTM Model Framework:

To get detailed evidence, we built a Neural Convolution network. The entire network is made up of three hidden layers. The Convolutional layer and a grouping layer are converged as the first 2 layers which comprises as hidden layers. The third layer is an output LSTM layer. With each hidden layer, the number of Convolutions is different. The network uses Convolution cores and shared cores to increase efficiency by constantly expanding the network configuration. The number of conversions should improve accuracy. We build 2 Convolution layers one with 64 filters and 128 filters on the other. Figure 2 is a paradigm for texture classification centered on the Convolutional Neural Network algorithm with the LSTM-70 as output. Throughout , it can be shown that the structure mainly consists of three phases:

Phase 1: Different Texture data set has a large amount of texture images that allows you to load datasets.
Phases 2: Training and feature extraction: the constructed CNN model is used for the training the texture dataset and automatically extracts features based on characteristics of the image.
Phases 3: Test: The SoftMax classifier is used to identify and collect the effects of the test.

3.TESTING THE MODELS

3.1.1 DATA-SET DESCRIPTION:

This section describes the details of the proposed approach and its working process. It also discusses the datasets used for the experiments. A brief description of the selected machine learning classifiers is provided as well.

3.1.2 DATA SETTING AND PREPROCESSING

Datasets used in the current study This study makes use of three different datasets to perform sentiment analysis. The purpose of using multiple datasets is to evaluate the performance of the proposed model on multifarious datasets to determine its efficacy. Each dataset contains a different number of Tweets and their associated classes. Table1describes the details of each dataset concerning the total numbers of records and classes. Each dataset contains a different number, as well as different types, of attributes, so it would be more appropriate to describe the details of the attributes used in this study before moving to data visualization.

TABLE 1:

Dataset name	**Total records**	**No. of classes**
Twitter airline sentiment (Dataset 1)	14 640	3
Women’s e-commerce clothing reviews (Dataset 2)	23 486	5
Twitter sentiment analysis for hatred speech detection (Dataset 3)	530	2

3.1.3Data pre-processing

The datasets are semi-structured/unstructured containing a large amount of unnecessary data, which plays no significant role in the prediction. Moreover, a large dataset requires a longer training time, and stop words reduce the accuracy of the prediction. Therefore, text pre-processing is required to both save computational resources and increase prediction accuracy. Text pre-processing[35] plays a vital role in more accurate prediction and elevates the model’s performance. The following steps are carried out in the pre-processing phase, as shown in Figure 4.

Tokenization: This involves the splitting of continuous text into words, symbols, and elements (called tokens).[36] It has a significant impact on the performance of the subsequent analysis, so it should be correct and efficient.[37]
Stop word removal: In the next step, stop words are removed from the Tweets. Although stop words make sentences more readable, they do not add value to text analysis. The removal of stop words increases the efficiency of the classification algorithm.[38]

Short word removal: Short words with a character length of less than three are removed from the Tweets. Research[39] finds out that SVC is not robust with short words and its accuracy is affected if tweets contain short words. Hence, short words are discarded to increase the robustness and efficacy of classifiers.

Case conversion: After short words are removed, the text in the Tweets is converted to lowercase. This is an important step because the analysis is case sensitive. The probabilistic models, forexample, consider "Bad," and "bad" as different words, and they count the occurrence of each word separately.[38] If the words are not converted to lowercase, it could impair the efficiency of the classifier.

Stemming: Stemming is the process of removing the affixes from words and restoring the words to their root forms.[40] For example, enjoys, enjoying, and enjoyed are variations of "enjoy"

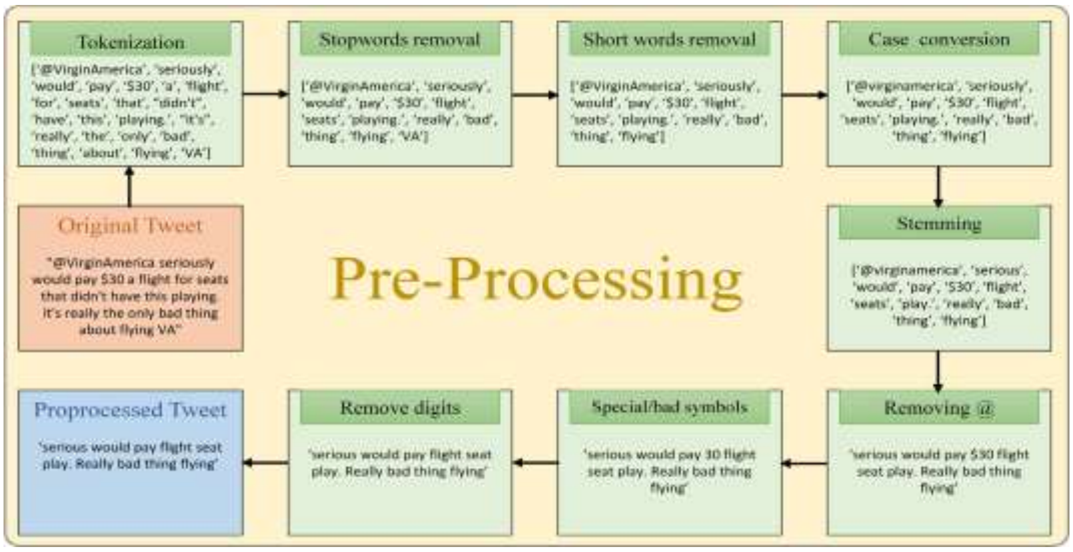

Tokenization
['@VirginAmerica', 'seriously', 'would', 'pay', '$30', 'a', 'flight', 'for', 'seats', 'that', "didn't", 'have', 'this', 'playing.', "it's", 'really', 'the', 'only', 'bad', 'thing', 'about', 'flying', 'VA']
Stopwords removal
['@VirginAmerica', 'seriously', 'would', 'pay', '$30', 'flight', 'seats', 'playing.', 'really', 'bad', 'thing', 'flying', 'VA']
Short words removal
['@VirginAmerica', 'seriously', 'would', 'pay', '$30', 'flight', 'seats', 'playing.', 'really', 'bad', 'thing', 'flying']
Case conversion
['@virginamerica', 'seriously', 'would', 'pay', '$30', 'flight', 'seats', 'playing.', 'really', 'bad', 'thing', 'flying']
Original Tweet
"@VirginAmerica seriously would pay $30 a flight for seats that didn't have this playing it's really the only bad thing about flying VA"
Pre-Processing
Stemming
['@virginamerica', 'serious', 'would', 'pay', '$30', 'flight', 'seats', 'play.', 'really', 'bad', 'thing', 'flying']
Proprocessed Tweet
'serious would pay flight seat play. Really bad thing flying'
Remove digits
'serious would pay flight seat play. Really bad thing flying'
Special/bad symbols
'serious would pay 30 flight seat play. Really bad thing flying'
Removing @
'serious would pay $30 flight seat play. Really bad thing flying'

FIGURE3.1: Steps carried out in data pre-processing

with the same meaning. Removing the suffixes helps reduce feature complexity and improves the learning capability of classifiers.

Removing @ and bad symbols: After stemming, words starting with are removed because Twitter assigns a unique name for each subscriber, which starts with . After that, special symbols are removed. This study found that a few symbols remain in the Tweets even after the specialsymbol–removal phase is complete. So the bad symbol step follows to remove symbols (eg, a heart).

As the next step, numeric values are removed from the Tweets because they do not possess any value for text analysis, and removing them decreases the complexity of the models' training. Table 4 shows a few Tweets before and after pre-processing has been performed.

4.RESULT OF DEEP NEURAL NETWORK

Deep LSTM and the CNN-LSTM were trained using word2vec features over 500 epochs. Results for model training and testing accuracy, and training and testing loss are shown in Figure 6. Accuracy results obtained using LSTM and CNN-LSTM for each dataset are shown in Table 11. Results show that the proposed model, which is based on the combination of CNN and LSTM networks, outperformed the LSTM classifier. The accuracy of CNN-LSTM was 0.82, compared to LSTM, which was able to achieve an accuracy of 0.76 on the Tweet datasets.

4.1Performance comparison of CNN-LSTM with machine learningclassifiers:

The performance of the proposed CNN-LSTM model was compared with selected machine learning classifiers. Since machine learning-based classifiers show better performance when used with TF-IDF features, their performance with TF-IDF was compared with the CNN-LSTM performance. the accuracy of the proposed model against the machine learning classifiers. Results prove that the proposed model outperforms other selected classifiersfor the three datasets used in the experiments. It performed better on Dataset 2, even whenother machine learning classifiers performed poorly and achieved an accuracy of 0.781. Results show that the proposed model performed much better on Dataset 3, and its accuracy was 0.920 while VD-CNN[13] and CNN-Bi-LSTM[13] had an average accuracy of 0.515 and 0.859, respectively, at the same time. Similarly, precision, recall, and F1-score were also high for the proposed model.

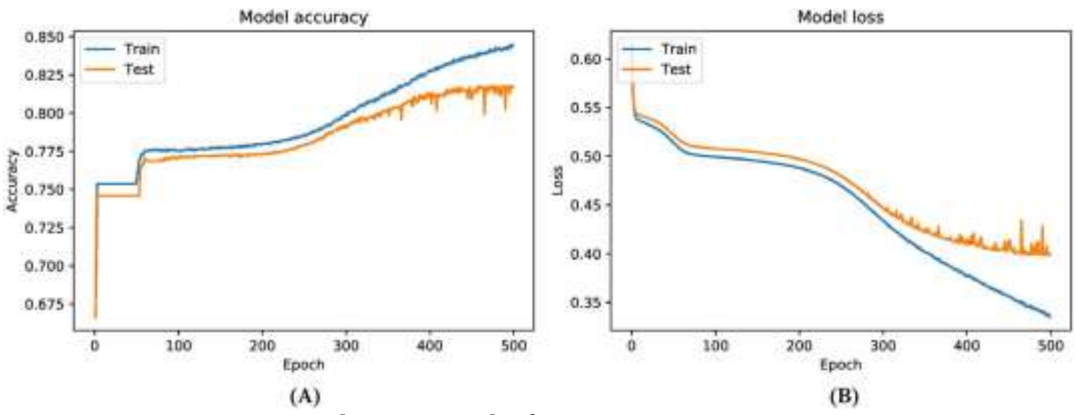
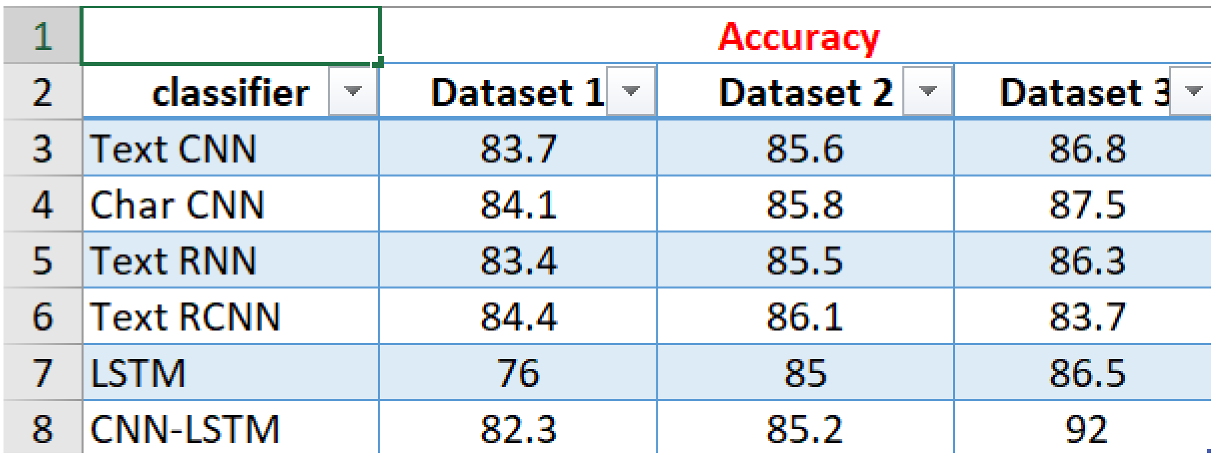
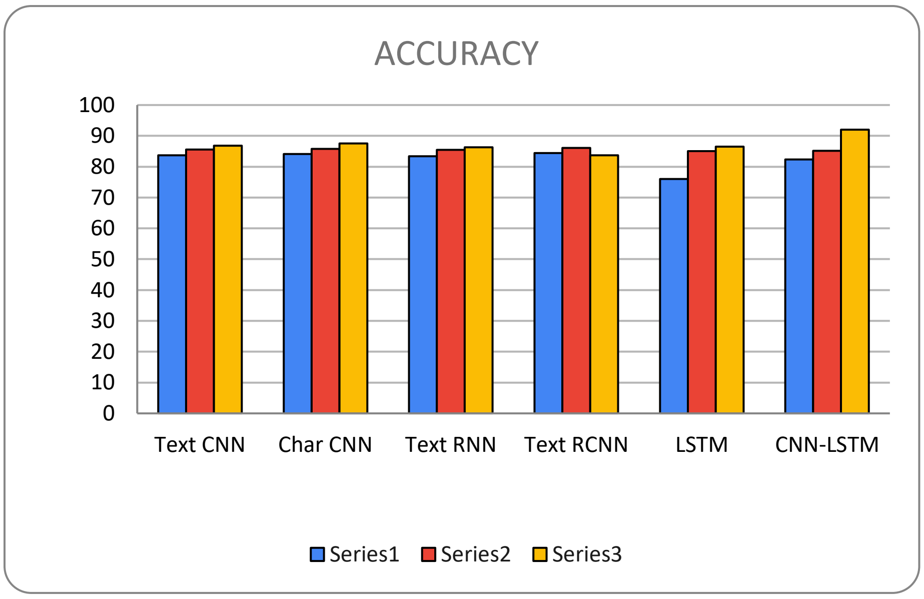


FIGURE 4.1 Training and testing results for CNN-LSTM. A, Accuracy. B, Loss



1		Accuracy		
2	classifier	Dataset 1	Dataset 2	Dataset 3
3	Text CNN	83.7	85.6	86.8
4	Char CNN	84.1	85.8	87.5
5	Text RNN	83.4	85.5	86.3
6	Text RCNN	84.4	86.1	83.7
7	LSTM	76	85	86.5
8	CNN-LSTM	82.3	85.2	92



5.CONCLUSION

In order to ensure the short-term and small-scale data requirements of sentiment classification in public opinion analysis scenarios, this paper proposes a model UCRNN that uses user attributes to expand the input information. It used CNN network to extract user attribute features and used Bi-LSTM to extract text data features, and the features were fused in parallel for emotion classification. Through the data obtained on social media Sina Weibo, the experiment proved that UCRNN can achieve the best classification result with less training time than Text RNN. UCRNN is a very effective solution for social media text emotion classification task, but there are still some problems. On the one hand, in terms of the selection of user attributes, some of the 12 user attributes selected in this paper have been confirmed by previous studies to be directly related to emotion categories, but some of them cannot be determined whether their addition has a positive or negative effect on the whole model. In the experiment, it is found many choices of user attributes may cause overfitting, and too few choices cannot greatly improve the classification effect. Further research will conduct a more in-depth analysis of this problem, select a more appropriate user attribute collocation and give the reasons. On the other hand, whether the selection of the network model and the integration of the dual model can be improved also needs further research. Using CNN to extract user attributes in UCRNN is a good method, but at the beginning of designing the model, a fully connected neural network was also considered. In the fusion part of the two models, this model uses the parallel splicing of feature vectors, and whether the use of weight distribution or serial connection can get better classification results still needs further verification. The success of UCRNN is mainly due to its use of multi-modal information, which can increase the amount of features contained in small-scale data. The use of multi-modal information will be a very challenging and meaningful research direction in the field of artificial intelligence, which mainly includes two points: firstly, the choice of multimodal information, which should have the characteristics of easy access and strong relevance; secondly, how to reduce the parameters of the model, because the amount of parameters of the neural network that extracts the multi-modal information features has a great impact on the training time.

REFERENCES:

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘‘BERT: Pre-training of deep bidirectional transformers for language understanding,’’ in Proc.NAACL-HLT, Minneapolis, MI, USA, Jun. 2019, pp. 4171–4186.
[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,L. Kaiser, and I. Polosukhin, ‘‘Attention is all you need,’’ in Proc. NIPS,Long Beach, CA, USA, Dec. 2017, pp. 5999–6009.
[3] G. Tang, M. Müller, A. Rios, and R. Sennrich, ‘‘Why self-attention?A targeted evaluation of neural machine translation architectures,’’ 2018,arXiv:1808.08946. [Online]. Available: https://arxiv.org/abs/1808.08946
[4] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le,‘‘XLNet: Generalized autoregressive pretraining for language understanding,’’ in Proc. NIPS, Vancouver, BC, Canada, Dec. 2019, pp. 1–18.
[5] Y. Kim, ‘‘Convolutional neural networks for sentence classification,’’ 2014, arXiv:1408.5882. [Online]. Available: https://arxiv.org/abs/1408.5882
[6] D.-G. Ko, S.-H. Song, K.-M. Kang, and S.-W. Han, ‘‘Convolutional neural networks for character-level classification,’’ IEIE Trans. Smart Process.Comput., vol. 6, no. 1, pp. 53–59, Feb. 2017.
[7] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, ‘‘Bag of tricks for efficient text classification,’’ 2016, arXiv:1607.01759. [Online]. Available:https://arxiv.org/abs/1607.01759
[8] F. Li, M. Zhang, G. Fu, T. Qian, and D. Ji, ‘‘A Bi-LSTM-RNN model for relation classification using low-cost sequence features,’’ 2016,arXiv:1608.07720. [Online]. Available: http://arxiv.org/abs/1608.07720
[9] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, ‘‘Convolutional recurrent neural networks for text classification,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Austin, TX, USA, Jan. 2019, pp. 2267–2273.
[10] X. Sun and J. He, ‘‘A novel approach to generate a large scale of supervised data for short text sentiment analysis,’’ Multimedia Tools Appl., vol. 79,no. 9, pp. 5439–5459, Feb. 2018, doi: 0.1007/s11042-018-5748-4.

[11] A. Kumar, K. Srinivasan, W.-H. Cheng, and A. Y. Zomaya, ''Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data,'' Inf. Process. Manage.vol. 57, no. 1, Jan. 2020, Art. no. 102141, doi: 10.1016/j.ipm.2019.102141.
[12] S. Bairavel and M. Krishnamurthy, ''Novel OGBEE-based feature selection and feature-level fusion with MLP neural network for social media multimodal sentiment analysis,'' Soft Comput., vol. 24, no. 24,pp. 18431–18445, Dec. 2020, doi: 10.1007/s00500-020-05049-6.
[13] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, ''Using linguistic cues for the automatic recognition of personality in conversation and text,''J. Artif. Intell. Res., vol. 30, pp. 457–500, Nov. 2007.
[14] L. Mitchell, M. R. Frank, K. D. Harris, P. S. Dodds, and C. M. Danforth,''The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place,'' PLoSONE, vol. 8, no. 5, May 2013, Art. no. e64417, doi: 10.1371/jour-nal.pone.0064417.
[15] X. Li, Z. Wang, C. Gao, and L. Shi, ''Reasoning human emotional responses from large-scale social and public media,'' Appl. Math. Comput.,vol. 310, pp. 182–193, Oct. 2017.