Sentiment Analysis with Novel GRU based Deep Learning Networks

Chary Vielma
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
chary.vielma@csu.fullerton.edu

Abhishek Verma
Department of Computer Science
California State University, Northridge
Northridge, CA, USA
abhishek.verma@csun.edu

Doina Bein
Department of Computer Science
California State University, Fullerton
Fullerton, CA, USA
dbein@fullerton.edu

Abstract— **Consumers make decisions online on diverse websites based on recommendations from people they have never met. This shared online input provides thus insight into the way they perceive things such as products, services, or events, can be beneficial to some degree to the users who are looking to make decisions. Sentiment identification is an ongoing research topic that has successfully shown results for movie and product reviews, blogs, social media post. As users, we shop for the best fit. Text classification, a subcategory of natural language processing, is the task of categorizing text to represent the word and its use in context. Neural networks have been widely used to implement text classification mechanisms for sentiment analysis. In this research we aim to use movie reviews from the internet movie database (IMDb) dataset [1] to perform neural network sentiment analysis. Two novel multi-branch models are explored: CNN-GRU and CNN-bidirectional GRU. The results show that while CNN-bidirectional GRU has slightly higher accuracy, the CNN-GRU has a comparable accuracy and did so with less training time.**

Keywords— GRU; Bidirectional GRU; sentiment analysis; natural language processing

I. Introduction

People have turned to the Internet in greater numbers than before to shop for products, services (such as car repairs, home maintenance, etc.), or even voting. Due to busy schedule or inconvenience, they rely more on the opinions expressed online, and in turn they themselves leave online comments. Users are producing more text to express in greater details their sentiment regarding something to be had, gained, or achieved. As the amount of online user input recorded across public forums accumulates, the need to interpret that information in a meaningful way has never been more important. Whether the objective is to collect feedback on products and services or to simply study the way people perceive the world around them, the field of Natural Language Processing (NLP) and principles of machine learning can be applied to explore such concepts. NLP refers to the use of computers to read and interpret words and their meaning from human languages. To represent language in terms that a computer can understand, words are given numerical values and treated with NLP techniques. The goal in our research is to categorize user opinions towards a topic as being positive or negative sentiments also known as sentiment analysis.

In sentiment analysis, the sentiment classification process can be done at the aspect level, sentence level, and document level [2]. In this research, to analyze the sentiment of one opinion in its entirety, each opinion is treated using the document-level classification approach. The context of words in an opinion depends on the ordering of the words themselves such as "That was a bad restaurant" has a different meaning than "That restaurant wasn't bad at all." Given a particular topic, identifying the polarity of someone's opinion, i.e. namely whether someone has positive, neutral, or negative opinions of it, has been traditionally the basis of sentiment analysis. A vast amount of online opinions is available online as blogs, public comments, social media sites, so the Internet has been the best source of such data. This might explain why sentiment analysis and opinion mining are often thought to be interchangeable terms. However, it is more accurate to describe sentiments expressed online as emotionally loaded opinions. Sentiment identification is an ongoing research topic that has successfully shown results for movie and product reviews, blogs, social media post. It has become very popular among researchers due to ease of using existing powerful and free software. It has become very popular with major companies too, interested in improving their products to attract more customers, bloggers to attract more readers, movies to attracter more watchers and increase revenue. Identifying the general attitude by analyzing the sentiment and opinions expressed could help companies identify who will support their products or services.

In the past, news stories have been traditionally neutral, and their purpose was to convey what has happened, has been observed, or has been stated by someone. These days, a significant amount of money can be gained or lost if certain news is made public. Fundamental analysis for example deals with the online public opinion of companies, and automated trading algorithms take into consideration the news and opinion stated by users online regarding the products or services offered by that company. Determining the sentiment in a text can be simply done by identifying and accounting the sentiment-bearing words by comparing with existing predefined collection of sentiment-bearing words. More complicated algorithms do not treat the sentiment-bearing words equally but consider the context of the words (words before and after) and adjust the weights depending on their position in the sentence. In our research, we used machine learning, or more specifically a convolutional neural network (CNN), to learn word association also known as word-level learning with word embeddings [3] for IMDb dataset. We use Gated Recurrent Unit (GRU) and bidirectional GRU layers.

The ensuing research is presented in five principal sections. Section II covers related and background work. Section III describes the IMDb dataset. Section IV details the methodology and experimental environment used to establish two proposed models and defines the functions of each layer in the network. Section V presents the results and discusses the findings. Section VI concludes our research presented in this paper.

II. Background and Related Work

Sentiment analysis in natural languages and movie reviews post have been two very popular topics among researchers and a lot of work has been done in document-level classification and contextual polarity disambiguation to some topic based on sentiment classification. Traditional machine learning classification methods have achieved some success, especially Support Vector Machines (SVM) and logistic regression [4]. Newer, non-neural network classification methods such as Maximum Entropy and Naive Bayes have also been used with some success for sentiment analysis. But in the recent years, convolutional neural networks have become very popular, because they do not depend on extensive manual feature engineering and extract the features automatically [5]. CNNs have shown good results for the sentiment classification of movie reviews which outperforms all the above-mentioned techniques. The authors of [6] introduced special type of neural network called Recurrent Neural Network (RNN), which maps phrases through word embeddings and a parse tree. The models used were based on changing the architecture of the RNN by adding Long Short-Term Memory (LSTM) units, Long Bidirectional Short-Term Memory (BiLSTM units).

A. Convolutional Neural Networks

A CNN is a type of deep neural network modeled after the neurons in the human brain. Advancements in computing capabilities have expanded the amount of processing that can be done with neural networks thereby more closely resembling the real-time human thought process. CNNs are comprised of fully connected layers and within each layer contain nodes. This architecture is meant to mimic the concept of neurons in the brain and the way in which sensory data is processed.

In order to train a CNN, data is given to an input layer. For each node in a layer, the output is multiplied by a weight and sent to every node in the next hidden layer. The forwarding of data across layers is what gives a CNN its feed-forward characteristic. A bias which is present at each neuron is added to that figure and is then processed through an activation function. This function uses the input at that node to decide whether that node is activated meaning an “on” state. This is similar to the sensory data in the brain which can trigger a neuron depending on what is being sensed. An output layer produces a predicted outcome based on the input provided by the nodes in the preceding hidden layer. It can then adjust the weights and biases to improve the accuracy based on the prediction. The dataset used for this type of analysis is labeled, meaning that each data sample is tagged with labels that are descriptive of the sample itself. For instance, an image of a car would be labeled as such and contain descriptors indicative of a car. This allows us to employ supervised learning in which inputs are mapped to outputs based on what the model is inferring about given inputs and in which categories the outputs best fit.

CNNs have been widely used for image recognition and classification due to the way in which inputs can be processed [7]. Because the inputs are a grid of pixels, the pixels can be broken down and analyzed in subsets sequentially. During this step, any relationships between the smallest units of measurement, which are pixels, are preserved in relation to one another. After processing at each node is complete, the results are stored in spatial matrices and sent to the next layer. Research shows that this method has turned out to be successful for text processing as well.

B. Gated Recurrent Units

While a CNN only has knowledge of its current state, the GRU architecture introduced in [8] maintains an internal memory state that is updated only when triggered by a signal. This gives states the ability to store dependencies from previous steps over longer periods of time. In the context of text processing, associations between words that have occurred over a longer period persist if their meaning is relevant [9].

An additional variation of a GRU that is used in our research is the novel bidirectional GRU with multiple branches. This more robust architecture takes the concept of the GRU internal memory state and applies it to both directions of a state. This allows the current state to receive information from a past and future state allowing the state to adapt at a faster rate [10].

C. Text Classification and Sentiment Analysis

Text classification is a widely studied problem due to all its valuable uses. The problem can be solved with vastly different approaches. Long short-term memory (LSTM) networks first introduced in [11] have been widely used for text classification because they address the need to have words take on different meanings depending on the words that come before and after it. It accomplishes this with use of an internal memory state. This state remembers long-term dependencies which are controlled by internal gates that decide whether to allow a new value to be added to a cell, erase the internal cell, how much to write to the cell, and what scale to use to decide how much of the cell to output. The work presented in [12] used LSTM as well as sentence fusion for natural language inference. They first started by creating word embedding vectors for two sentences and performing sentence embedding to produce two-sentence vectors. Sentence fusion occurs when these sentence vectors are processed with multiplication and subtraction operations on the vector data. Finally, their results are categorized. This allows them to make inferences such as King – man + woman = Queen [12].

In [13] the research used tweets to create embeddings which could then be classified. Their classification process used both semantics and sentiment of each tweet. The categories were positive, negative, and neutral. Another similar study in [14] used social media posts to create word embeddings which they could then input to a bidirectional GRU. The output of the layer was then passed to a convolutional neural network (CNN). The result of this model was a prediction into a user's interests. In [15] they used a combination of Latent Dirichlet Allocation text representation and a CNN for sentiment analysis of online user input. This method of text representation uses a distribution system to categorize text as topics. This work is unique in that they process text with the Gibbs sampling method to calculate the probability that a word belongs to the topic.

Neural networks can also be used to determine someone's personality as was shown in [16]. As they pointed out, what a person writes and the way they express themselves is a better metric to determine their personality than if you were to ask them to describe themselves. They first create sentence vectors with a forward-sending GRU encoder. The sentences are embedded and processed by a document encoder. The document-level processing is done with a CNN.

The research presented in [17] sought out to create a model that could detect hate speech online. They identified and grouped words they referred to as 'othering' to detect when a comment was trying to exclude a subset of individuals from a group such as 'We want you gone.' The 'we' and 'you' served as indicators of divisive language. From this they were able to classify comments into appropriate categories. They also used word and sentence embeddings to process the data. They attempted several approaches to test their 'othering' system some of which include support vector machines, logistic regression, gradient boosted decision trees and LSTMs, CNNs, and GRUs. Research presented in [26], [27] uses deep convolutional neural networks on text data for classifying sincerity and text summarization.

Using online user input data serves various purposes in business such as a marketing team looking to determine what content to advertise to certain users or gather feedback on a newly-released product. While there are many benefits in researching sentiment analysis, there are also countless examples of ways in which social media can be used in negative ways. From a societal standpoint, sentiment analysis is also important as it plays a role in studying society's behavior online. In our research we aim to create a model that can determine a review's polarity using the IMDb dataset. This model could also be used for other text datasets that look to determine the user's sentiment on a topic.

III. Dataset Description

The Association for Computational Linguistics' dataset used in our research is from the popular IMDb website. This dataset was first presented in [18] to study word vectors. The dataset is comprised of movie reviews collected from the IMDb website where users rate a movie on a scale of 1 to 10 stars where 10 is most favorable. All positive examples consist of ratings between 7 and 10, while all negative examples consist of ratings between 1 and 4. Neutral ratings of 5 and 6 are excluded. Users are also able to write text reviews for movies. Each movie contains at most 30 text reviews and each review has an average of 234.76 words with a standard deviation of 172.91 words. Because some reviews can be significantly longer, we limit a review to 500 or 550 words. Certain punctuation symbols and even emoticons such as ":(" are also included as they offer a different type of insight into the polarity of the review. In some ways, an emoticon can be more expressive and a better indicator of a person's opinion as it is not as ambiguous as words can sometimes be.

The dataset offers a total of 100,000 text-based movie reviews, half of which are unlabeled reviews for testing. The other half of the reviews are labeled with a 1 or 0, denoting positive or negative respectively. The labeled reviews are half positive and half negative to introduce an equal number of reviews for each classifier.

The Python deep learning library Keras is a high-level Tensorflow API [19][20]. Keras performs text preprocessing on the IMDb dataset in order to produce an ordered list of a maximum dictionary length. The indices of the words in a review are what is sorted in a dictionary by frequency from highest to lowest. The maximum dictionary length is a parameter that limits the top number of frequent words to consider. For instance, if a word occurs once in the entire database, there would be no need to include it in training. The maximum sequence length refers to the parameter that limits the number of words in a review. A review is padded with zeros for reviews with shorter review sequences and is truncated to the maximum sequence length for longer reviews. The vector length refers to the dimension of the vector in which to embed each word during the embedding process.

IV. Proposed Multiple Branches CNN-GRU and CNN-Bidirectional GRU Models

Training of our model was conducted on Ubuntu server. The hardware included an Intel Xeon E5-2630 v4 @ 2.20GHz CPU and a SuperMicro GPU 7x GTX 1080 Ti graphics card. To train the model, the API library Keras was used with Python 2.7. The Keras library works as a wrapper for the powerful machine learning platform TensorFlow. The experiments were conducted with Keras version 2.0.0 and TensorFlow version 1.0.1. This provides software compatibility between the two libraries.

The proposed models discussed in this research use a 1-dimensional CNN layer across multiple branches which are fed into a GRU or bidirectional GRU layer. We compare our findings in the Discussion section with the CNN-LSTM model presented in [21].

In Figure 1 we present the CNN-bidirectional GRU branch of the proposed multi-branch CNN-bidirectional GRU model 33. It uses a bidirectional GRU layer whose internal architecture resembles that of an LSTM except that a GRU only has two gates instead of three like the LSTM [22]. LSTM and GRU have many similarities but performance could vary depending on the dataset. The following subsections detail the layers in the network diagram seen in Figure 1.

A. Convolution

Each branch of the model receives an embedding layer as an input and produces a tensor of outputs. During this step, the kernel has a shape of *kernel size * size of embedding*. The kernel size refers to the number of consecutive words that will be studied. This is how the model learns the context of words such

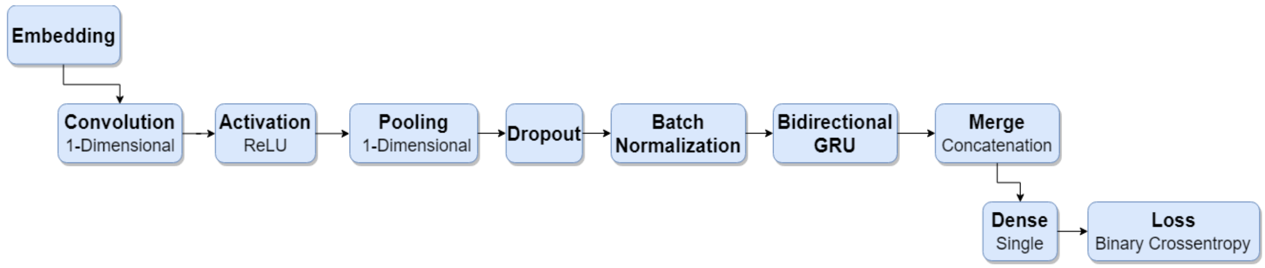

Fig. 1. CNN-Bidirectional GRU branch diagram of proposed multi-branch model 33.

as the difference between "Bad movie" and "Not a bad movie at all." In this example, the model must be able to discern that if the word "not" precedes the word "bad," that this has a different connotation than if "bad" is used by itself. The number of filters used in this layer is 128 units.

B. *Activation*

The outputs of the convolution layer receive a rectified linear unit (ReLU) activation. By performing a transformation of the data, we expand the model's learning capabilities by adding biases to the network.

C. *Max Pooling*

The 1-dimensional max pooling of each branch reduces the input's dimensionality and the amount of computational resources needed by scaling the sizes down to manageable ranges. These methods reduce the amount overfitting that occurs in the model. The output shape of each layer is dependent on the kernel size.

D. *Branch Dropout*

The dropout layer overwrites a randomly selected number of inputs to 0. This value is given by the dropout parameter. It prevents the model from overfitting by ensuring it does not memorize any pieces of information. The dropout rates used for testing were between the ranges of 0.4 to 0.6 and were adjusted as merge dropout rates were adjusted as well. When both values were too high, it caused the model to underfit as too many pieces of relevant information were omitted.

TABLE I. INPUT DATA LENGTHS

Proposed Models	Dictionary (words)	Sequence (words)	Vector size
Model 25	5,000	500	32
Model 33	5,000	500	32
Model 42	5,500	550	64

TABLE II. NETWORK LAYER CONFIGURATION OF TOP THREE PERFORMING MODELS

Proposed Models		**Model 25**	**Model 33**	**Model 42**
Convolution	Four branches and their Kernel Size	3/5/7/9	3/5/7/9	3/5/7/9
	Filters	128	128	256
	Kernel Regularizer	L2(0.01)	L2(0.01)	L2(0.01)
Activation - Type		ReLU	ReLU	ReLU
Max Pooling - Pool Size		2	2	2
Branch Dropout - Rate		0.4	0.4	0.4
Batch Normalization - Present		Yes	Yes	Yes
Type - Units		GRU(128)	Bidirectional GRU(128)	Bidirectional GRU(128)
Merge Dropout - Rate		0.1	0.1	0.1
Optimizer	Type	RMSprop	RMSprop	RMSprop
	Learning Rate	0.01	0.01	0.01
	Learning Rate Decay	0.1	0.1	0.1
Accuracy (Maximum) [0-1]		0.897	0.898	**0.900**

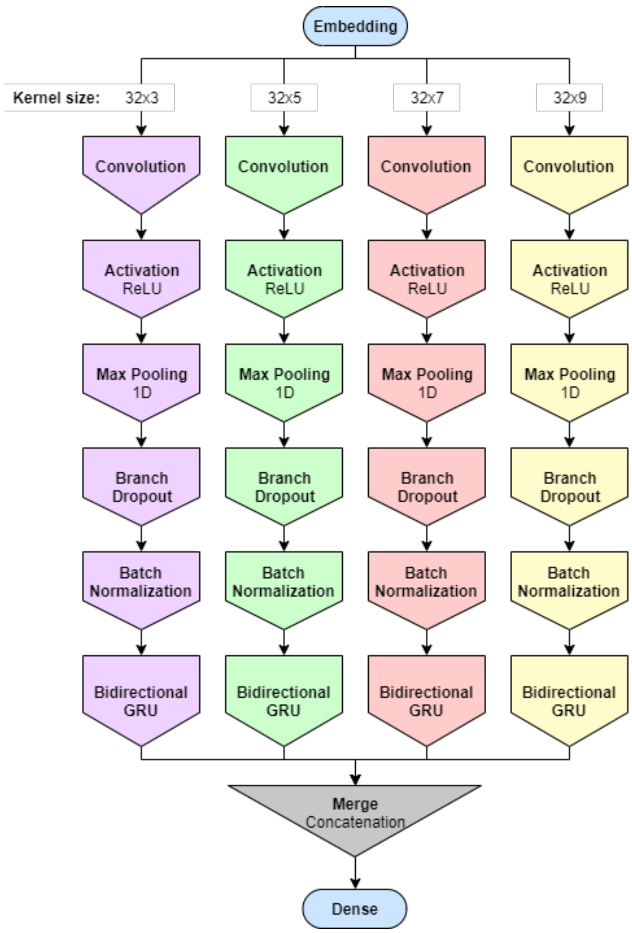

Fig. 2. Layered diagram of proposed multi-branch CNN-bidirectional GRU model 33. The embedding layer is the input data to multiple branches in the second tier. After the branches are concatenated in the Merge layer, the Dense layer receives a single input.

E. Batch Normalization

In order to reduce internal covariate shift, a batch normalization layer is used. This normalizes the batch distribution of values at the branch level.

F. Gated Recurrent Unit (GRU)

Each branch goes through either a GRU or bidirectional GRU layer which is the part of the model which allows us to store information from a different state in a memory cell. The bidirectional mechanism allows us to pass information forwards and backwards from other states across layers. The light-weight GRU architecture was chosen because of its improvements over other neural networks such as its predecessor LSTM.

G. Concatenation

Given a set of tensors as an input and an axis to concatenate along, the concatenation layer returns one tensor containing the original input size. After this operation, there is only one branch again.

TABLE III. ACCURACY COMPARISON

Proposed Models	**Type**	**Accuracy [0-1]**
Model 25	CNN-GRU	0.897
Model 33	CNN-Bidirectional GRU	0.898
Model 42	CNN-Bidirectional GRU	**0.900**
Model in [21]	CNN-LSTM	0.895

H. Dense

The dense layer performs matrix multiplication This step is followed by a sigmoid activation function to conform the output between 0 and 1. This results in a single output.

I. Loss Function and Optimizer

The binary cross entropy loss function is used for this model. Loss is calculated using two classes, 0 and 1, where 0 denotes a negative review and 1 denotes a positive review. The loss is calculated on the single and final output of the dense layer.

Adam, RMSprop, and Stochastic Gradient Descent (SGD) optimizers were used during testing. Testing was conducted with varying combinations of learning rates and learning rate decay parameters. The best-performing optimizer was RMSprop with a learning rate of 0.01 and learning rate decay of 0.1.

Figure 2 shows the multi-branch CNN-bidirectional GRU model. Four branches are used with the following kernel sizes: 32x3, 32x5, 32x7, 32x9 where 32 represents the embedding size and 3, 5, 7, and 9 represent the phrase sizes. By using multiple branch sizes, the model learns to review and understand short phrases of a given size; in this case, phrases of word size 3, 5, 7, and 9. It also learns to interpret the context in which words are used.

Table II details the parameters given to each layer for the top 3 performing models. As highlighted in Table I, the dictionary, sequence, and vector length parameters for proposed multi branch CNN-bidirectional GRU model 42 are larger than for proposed CNN-GRU model 25 and CNN-bidirectional GRU model 33.

V. RESULTS AND DISCUSSION

As outlined in Table III, the CNN-GRU model 25 yielded an accuracy of 89.7% and the CNN-Bidirectional GRU model 33 produced an accuracy of 89.8%.

Although one model used bidirectional and the other did not, the parameters for both were the same as seen in Table II. The research conducted in [21] reported an accuracy of 89.5% which is within 0.2% of model 25 and 0.3% of model 33 and 0.5% of

TABLE IV. MODEL TRAINING TIME

Proposed Models	**Type**	**Time**
Model 25	CNN-GRU	**1 hr 54 mins**
Model 33	CNN-Bidirectional GRU	3 hrs 11 mins
Model in [21]	CNN-LSTM	2hrs 31 mins

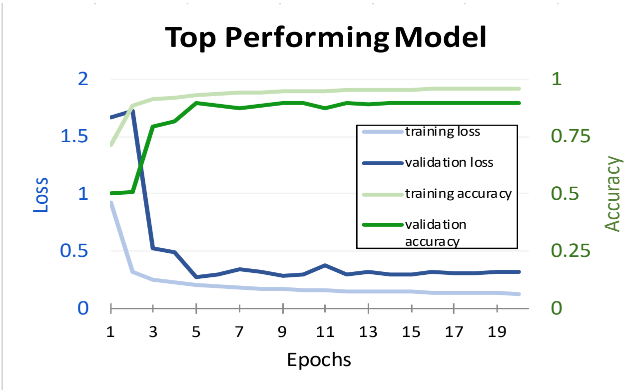

Fig. 3. Graph of proposed model 33 with dictionary length = **5000** words, sequence length = **500** words, and vector length = **32**.

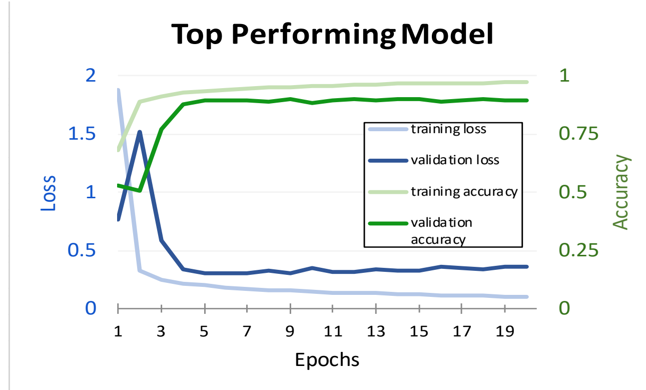

Fig. 4. Graph of proposed model 42 with dictionary = length **5500**, sequence length = **550**, and vector length = **64**.

Model 42. GRU models are slightly more accurate than the LSTM in [21].

Figure 3 shows the performance of model 33 across 20 epochs of training and validation. We can observe that between epochs 4 and 5 there was a sudden increase in validation accuracy and decrease in validation loss. Similarly, between epochs 10 and 11 there was a subtle drop in validation accuracy and an increase in validation loss. The model stabilized at around the 13th epoch.

The CNN-Bidirectional GRU model 42 yielded 90.0% accuracy. Figure 4 shows the performance of this model across 20 epochs of the training and validation results. The training loss for this model steadily decreased while the validation loss began increasing after the 11th epoch. After this epoch, the training accuracy and validation accuracy continued to steadily increase.

Table IV shows the training time for our top 2 performing models as well as the runtime for the model presented in [21]. CNN-GRU model is the best choice because while it still offers comparable accuracies, it does so with a faster training time. This makes the CNN-GRU an excellent choice for a neural network to use for text processing. The high runtime of CNN-Bidirectional GRU model is attributed to the additional bidirectional processing.

VI. Conclusion

The research presented in this paper aimed to explore the use of GRU and bidirectional GRU architectures in combination with CNN layer for text classification; more specifically analyzing sentiments with the IMDb dataset. We proposed multi branch CNN-GRU and CNN-Bidirectional GRU networks. Proposed models perform better in terms of accuracy in comparison to previously published model. In closing, the research done in this paper contributes to the fields of text classification and neural networks by further expanding our knowledge and proposing usable models to conduct sentiment analysis with user text.

References

[1] A. L. Maas, R. E. Daly, P T. Pham, D. Huang, A. Y. Ng, and C. Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

[2] J. Mir and M. Usman, "An effective model for aspect based opinion mining for social reviews," 10th Int. Conf. Digit. Inf. Manag. ICDIM 2015, no. Icdim, pp. 49–56, 2016.

[3] D. Xue et al., "Deep learning-based personality recognition from text posts of online social networks," Appl. Intell., vol. 48, no. 11, pp. 4232–4246, 2018.

[4] I.K. Chowdhury, S. Latif, and Md. S. Hossain, "Sentiment Intensity Analysis of Informal Texts" International Journal of Computer Applications, vol. 147, pp. 24-31, August 2016

[5] K. Kowalska, D. Cai, and S. Wade. "Sentiment Analysis Of Polish Texts" International Journal of Computer and Communication Engineering, vol. 1, pp. 39-42, May 2012

[6] R. Socher, C.C.-Y. Lin, A.Y. Ng, and C. D. Manning, "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," in Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, June 28 - July 2, 2011, L. Getoor, T Scheffer, Eds. USA: Omnipress, 2011. pp. 129-136

[7] C. Szegedy et al., "Going deeper with convolutions," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June, pp. 1–9, 2015.

[8] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., pp. 1724–1734, 2014.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 3, pp. 2067–2075, 2015.

[10] T. Liu, S. Yu, B. Xu, and H. Yin, "Recurrent networks with attention and convolutional networks for sentence representation and classification," Appl. Intell., vol. 48, no. 10, pp. 3797–3806, 2018.

[11] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. Neural computation, 9(8):1735– 1780.

[12] S. Zhang, S. Liu, and M. Liu, "Natural language inference using LSTM model with sentence fusion," Chinese Control Conf. CCC, pp. 11081–11085, 2017.

[13] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder," SIGIR 2016 - Proc. 39th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 1041–1044, 2016.

[14] J. Kang, H. S. Choi, and H. Lee, "Deep recurrent convolutional networks for inferring user interests from social media," J. Intell. Inf. Syst., vol. 52, no. 1, pp. 191–209, 2019.

[15] L. xia Luo, "Network text sentiment analysis method combining LDA text representation and GRU-CNN," Pers. Ubiquitous Comput., pp. 405–412, 2018.

[16] D. Xue et al., "Deep learning-based personality recognition from text posts of online social networks," Appl. Intell., vol. 48, no. 11, pp. 4232–4246, 2018.

[17] W. Alorainy, P. Burnap, H. Liu, M. Williams: The enemy among us: Detecting hate speech with threats based ’othering’ language embeddings. arXiv preprint arXiv:1801.07495 (2018)

[18] S. Li, S. Yat M. Lee, Y. Chen, C. Huang, and G. Zhou. 2010. Sentiment classification and polarity shifting. In Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 635-643.

[19] F. Chollet and others, Keras. 2015 https://keras.io.

[20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[21] A. Yenter and A. Verma, “Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis,” 2017 IEEE 8th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2017, vol. 2018-Janua, pp. 540–546, 2018.

[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". arXiv:1406.1078

[23] M. Khodabakhsh, M. Kahani, and E. Bagheri, “Predicting future personal life events on twitter via recurrent neural networks,” J. Intell. Inf. Syst., 2018.

[24] T. Liu, S. Yu, B. Xu, and H. Yin, “Recurrent networks with attention and convolutional networks for sentence representation and classification,” Appl. Intell., vol. 48, no. 10, pp. 3797–3806, 2018.

[25] S. Wen and J. Li, “Recurrent convolutional neural network with attention for twitter and yelp sentiment classification arc model for sentiment classification,” ACM Int. Conf. Proceeding Ser., 2018.

[26] R. Chittari, D. Bein, A. Verma, et al., "Classifying Sincerity Using Machine Learning," the 19th International Conference on Information Technology: New Generations, Apr. 11-12, 2022, Virtual.

[27] A. Dingare, D. Bein, W. Bein, and A. Verma, "Abstractive Text Summarization using Machine Learning," the 19th International Conference on Information Technology: New Generations, Apr. 11-12, 2022, Virtual.