A Hybrid Deep Learning Model for Long-Term Sentiment Classification

Tapas Guha*
Department of CSE, Presidency University, Bengaluru, India.
E-mail: tapasguha@presidencyuniversity.in

K.G. Mohan
Department of CSE, Presidency University, Bengaluru, India.
E-mail: mohankg@presidencyuniversity.in

Received July 20, 2020; Accepted September 28, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17059

Abstract

With the omnipresence of user feedbacks in social media, mining of relevant opinion and extracting the underlying sentiment to analyze synthetic emotion towards a specific product, person, topic or event has become a vast domain of research in recent times. A thorough survey of the early unimodal and multimodal sentiment classification approaches reveals that researchers mostly relied on either corpus based techniques or those based on machine learning algorithms. Lately, Deep learning models progressed profoundly in the area of image processing. This success has been efficiently directed towards enhancements in sentiment categorization. A hybrid deep learning model consisting of Convolutional Neural Network (CNN) and stacked bidirectional Long Short Term Memory (BiLSTM) over pre-trained word vectors is proposed in this paper to achieve long-term sentiment analysis. This work experiments with various hyperparameters and optimization techniques to make the model get rid of overfitting and to achieve optimal performance. It has been validated on two standard sentiment datasets, Stanford Large Movie Review (IMDB) and Stanford Sentiment Treebank2 Dataset (SST2). It achieves a competitive advantage over other models like CNN, LSTM and ensemble of CNN-LSTM by attaining better accuracy and also produces high F measure.

Keywords

Sentiment Analysis, Deep Learning, Word2vec, CNN, BiLSTM, IMDB, SST2.

Introduction

With the dramatic overload of information in social media, substantial data containing opinions towards products, services, people and events is omnipresent in multiple modalities like text, audio and video in the form of comments, reviews, discussion forums, blogs, vlogs etc. Some form of emotion is attached with all these opinions. The primary challenge of a machine is to interpret these underlying sentiments and classify their polarity like positive, negative or neutral. Also polarity scores can be assigned to specify the intensity of the sentiment. The act of finding an optimally efficient approach to mine and analyze these data is a primary research area involving variety of domains like Natural Language Processing, Machine Learning, Deep Learning, Image Processing etc., which is known as Sentiment Analysis. This also includes discovering troll accounts on a social network and sarcasm detection in a voice. All this data can be utilized in socio-economic, business and political purposes, making sentiment analysis a highly significant feedback tool.

In language modeling and text processing, traditional methods like bag-of-word (BoW) and the ngram models had some disadvantages like semantic disconnection and data sparsity problem respectively. Then, linear classifiers like logistic regression gained prominence but failed to correlate features and classes. The emergence of multilayer neural networks and word embeddings resolved these issues. Since similar words are supposed to be closely placed in the vector space low dimensional real vectors are used to represent words to preserve correlation.

With the growing prominence of deep learning in recent times, alongside the massive availability of labeled data, deep neural models have supplanted a significant number of the traditional approaches used to tackle different NLP and machine vision tasks. These deep neural models have achieved top-notch performance on a variety of NLP tasks like language translation, automatic text summarization, name entity recognition and sentiment analysis.

Convolutional neural networks (CNN), recurrent neural networks (RNN) and few variations of the latter like Long Short Term Memory (LSTM), Gated recurrent Units (GRU) etc. are some of the deep learning models extensively utilized in emotion categorization. CNN being proposed in 1998, has solid adaptability and is extremely capable of pulling out local features from text while considerably getting rid of computational complexity and the number of training parameters due to its distinctive organization [1]. One landmark model proposed dynamic CNN for sentence modeling, which is able to deal with short as well as long-range local associations [2]. The other popular variant, RNNs can learn long-term dependencies and take care of sequential data by considering the current as well as the previous hidden layer output [3]. The standard RNN suffers from issues like vanishing and exploding gradient which was resolved by LSTM [4]. Another prominent approach employs a hierarchical document processing extracts feature vectors using a bidirectional recurrent layer [5]. One popular model for classification of Imagenet images successfully merged both deep convolutional and recurrent layers to achieve excellent results [6]. Another hybrid approach used a similar architecture for large scale speech recognition [7].This work proposes a hybrid model consisting of CNN and stacked BiLSTM over pre-trained word vectors to achieve long term sentiment analysis.

The remaining paper is arranged in five main portions. In section 2, all related sentiment classification approaches are reviewed. Section 3 describes the components of the proposed model and its detailed architecture. In section 4, the datasets used, model hyperparameters and optimization techniques applied are discussed in depth. Section 5 explains the model performance. Finally, section 6 concludes this paper and discusses some possible enhancements.

Related Works

The beginning of the 21st century saw the increasing popularity of Sentiment analysis and opinion mining. Initially, Naïve Bayes classifiers and Support vector machines (SVMs) were used to collect general sentiment from movie reviews in one prominent approach [8]. Another work proposed sentence level sentiment extraction using a knowledge corpus to map equivalent emotions from various setups [9].

With the increasing research and utilization of word embedding and predefined word vectors, sentence representation through semantic orientation using CNNs, RNNs, LSTMs and some hybrid neural net models has been implemented in a chain of methods. A simple neural network having a single one dimensional convolution layer with multi-width filters, a pooling layer for prominent feature extraction and a final fully connected layer produced considerable good performance in sentiment categorization [10]. Further, bag-of-word model used for distributed representations of words and phrases, has been particularly useful in text classification [11]. Extensive feature mining and document level sentiment categorization have been jointly implemented in few noteworthy deep neural network methods [12]. Multiple layers of

CNN have been utilized in [13] to keep track of long term dependencies which gets critical as the length of the input data increases.

Recursive neural networks, RNNs and LSTMs have been explored by numerous researchers because of their capability to deal with sequential data. One prominent approach proposed a new hybrid architecture named RNN Encoder-Decoder by merging auto-encoders and recurrent neural nets which produced good performance in capturing all relevant representation in statistical language transformation [14]. Another alternative of recursive neural network which effectively analyzes phrase level sentiment utilized vectored matrix to represent each node of the recursive [15]. Further, another reformation of the recursive network that uses each LSTM cell to capture information present in every child node improves semantic representation [16].

In line with the already mentioned approaches, a plethora of scientists have applied deep learning techniques to analyze synthetic emotion recently. One prominent work trained labeled data to classify sentiment using transfer learning [17]. Another approach proposed hierarchical deep learning for sentiment analysis at feature level [18]. One standout model utilized ensemble techniques in order to enhance deep sentiment analysis [27]. A thorough survey of sentiment classification using deep neural networks is offered in [28].

Standard recurrent neural networks which predicts depending only on the past word information falls short in some cases where tagging both the past and future word of a sentence is required [19]. Here comes the significance of bidirectional recurrent neural network and LSTM. One landmark approach proposed a hybrid deep learning model by combining CNN and bidirectional RNN for efficient character-level text categorization [20]. However, the performance of this model is affected if the number of available classes is reduced.

Proposed Architecture

The proposed framework is based on a hybrid deep learning model consisting of CNN and stacked bidirectional LSTM (BiLSTM) over pre-trained word vectors created by training Google News containing 100 billion words from. The multiple feature maps of CNNs have been found to be really capable of capturing the local correlation in the review data.

Usually, different word embeddings are amassed to generate a two-dimensional array, and then convolution filters of variable sizes are associated with a pre-decided frame of words to represent certain prominent features. After that, one among different types of feature pooling (generally max-pooling) is applied on new features, which are then integrated to frame the concealed depiction. Finally, one or more fully connected dense classification layer with sigmoid or softmax activation function produces the required prediction.

On the other hand, LSTM, the popular variant of RNN is efficient in keeping track of long term information. This work, being interested in capturing time-based information in both directions employs a variation of LSTM called Bidirectional-LSTM (BiLSTM). Since this particular deep neural network focuses on the past as well as the future information, it offers added context to the model and optimally improved learning is achieved [21].

1) Word-Level Embedding

Word embeddings or pre-trained word vectors are particularly significant as they aid in catching syntactic and semantic information related to the associated sentiment of the relevant context. Also, different types of classifications can be performed vectors which are also feature extractors. This model has used word2vec word embeddings that used Continuous Bag of Words (CBOW) and now employs skip-gram models in order to estimate the required vector representations of review words. These were pre-trained on Google News containing 100 billion words from.

2) Convolution Layer

The Google word2vec toolkit is utilized in constructing the word vectors, and then each sentence is represented by placing these word vectors one after another. The word2vec model is a variation efficiently used with the CNN architecture of [22]. Let $x_i \in \mathbb{R}^k$ be the k-dimensional word vector associated with the i-th word in an n padded sentence, which is represented as in (1).

$$x_{i:n} = x_1 \oplus x_2 \oplus x_3 \qquad (1)$$

Where: $\oplus$ is the concatenation operator, a convolutional operational constitutes a filter $w \in \mathbb{R}^{hk}$ which is used to produce a new feature when associated with a window of h words. For instance, a frame of words $x_{i:i+h-1}$, creates a feature c_i as shown in (2).

$$c_i = f(w \cdot x_{i:i+h-1} + b) \qquad (2)$$

Where: f is a nonlinear activation function like rectifier and $b \in \mathbb{R}$ is the bias value.

This is repeated in each consecutive time step of a given input sequence $\{x_{1:h}, x_{2:h+1}, \dots, x_{n-h\_1:n}\}$ to generate a feature map as shown in (3)

$$c = [c_1, c_2, \dots, c_{n-h+1}] \quad (3)$$

3) Long Short Term Memory

RNN utilize sequential information to produce the output depending on the previous computation. RNNs can be thought of as having some form of a memory, which captures long-term sequential information, as equated in (4)

$$h_t = f(x_t, h_{t-1}) \quad (4)$$

Where: $x_t \in \mathbb{R}^d$ is one time step from the input sequence $(x_1, x_2, \dots, x_t)$. $h_0 \in \mathbb{R}^d$,

often initialized as an all-zero.

It is noteworthy that the vital emotion components are not guaranteed to be present at the end, but may span across an entire sentence or a paragraph of a document. Since information from earlier words are given less importance than that from recent words, RNNs turn out to be a biased and low performing model in such sentiment analysis cases, obviously diminishing the effectiveness in keeping track of semantic context of the entire document. The naive recursive function of RNNs suffers from problem of vanishing gradient and exploding gradient [23]. To tackle these issues, one variant of the RNN, the LSTM model was initially proposed in. It was further modified in [24]. It has been attested to be really proficient in catching long-term associations in variable length sequence [25].

In order to work with the input vectors, LSTM recursively executes the cell blocks depending on both the earlier hidden state h_{t-1} as well as the current input x_t, where t is the current time and $t-1$ refers to the former time. In cell block, let i_t be the input gate, let f_t stands for the forget gate, let o_t be the output gate, let c_t represents the current memory cell and h_t be the current hidden state. In each timestep, the memory cell layers are updated as shown in (5) to (10).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (5)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{6}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7}$$

$$u_t = tanh\,(W_u x_t + U_u h_{t-1} + b_u) \tag{8}$$

$$c_t = i_t * u_t + f_t * c_{t-1} \tag{9}$$
$$h_t = o_t * tanh\,(c_t) \tag{10}$$

Where: σ refers to the logistic sigmoid function, $W_i, W_f, W_0, W_u, U_i, U_f, U_0$ and U_u are weights, and b_i, b_f, b_o, b_u, are bias vectors.

4) Bidirectional LSTM

Since this proposed model intends to capture time-based information in both directions, it employs a variation of LSTM called BiLSTM which specifically serves this purpose. Fig. 1 illustrates the architecture of a standard BiLSTM.

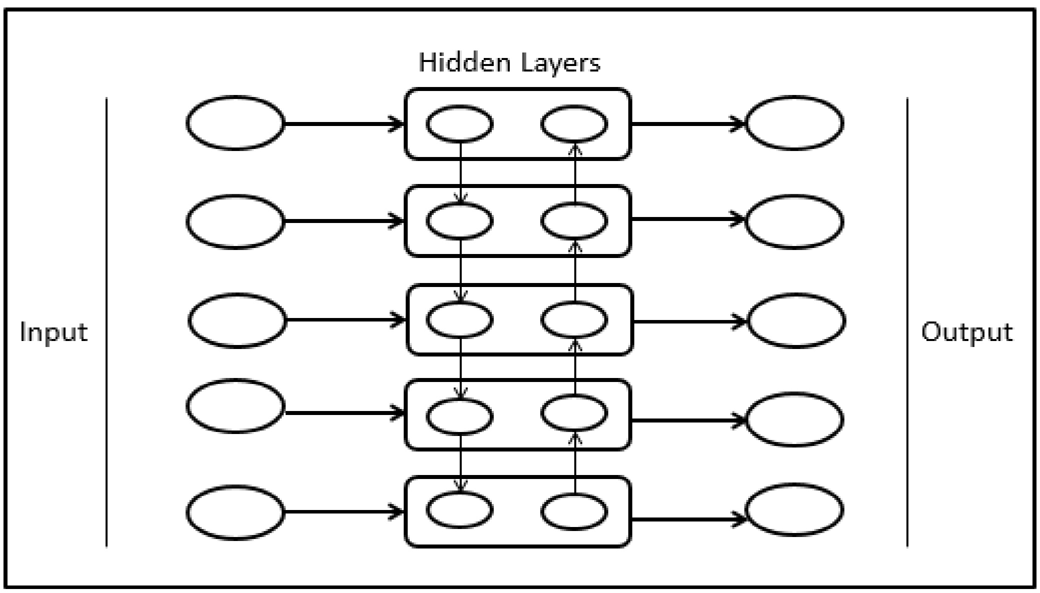

Fig. 1 Architecture of standard BiLSTM [42]

In BiLSTM, two hidden layers are trained over the same input sequence, one on the unchanged input series, and the second on the inverted duplicate of the same input series. The goal is to associate both the hidden layers of inverse directions to the same desired output.

In the proposed model, the embedded sequence is transformed into a series of real-valued dense vectors, then a one dimensional convolutional layer is employed to yield an undersized sequence of feature vector, as shown in (11).

$$FV = (fv_1, fv_2, \dots, fv_T) \tag{11}$$

This generated vector is then supplied to a BiLSTM layer, producing a pair of reverse sequence vectors as in (12) and (13).

$$H_f = (\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_T}) \tag{12}$$

$$H_r = (\overleftarrow{h_1}, \overleftarrow{h_2}, \dots, \overleftarrow{h_T}) \tag{13}$$

Where: H_f and H_r are the forward and reverse vectors respectively. The final hidden states of both directions are concatenated to produce a fixed-dimensional vector as shown in (14).

$$h_{fd} = [\overrightarrow{h_T}; \overleftarrow{\mathrm{h}_1}] \tag{14}$$

Finally, this is sent to a fully connected dense layer to calculate the predictive probabilities of all the available input categories.

5) Sentiment Categorization

Finally, this model has a fully connected dense layer which takes these deep sentence representations as input categorization of texts. This classification layer transforms this fixed-dimensional input from previous layer and then a softmax activation function is applied to compute the predictive probabilities for all the classes, finally generating a single output, as shown in (15).

$$p(y = k|X) = \frac{\exp(W_K^T X + b_k)}{\sum_{k'=1}^{k} \exp(W_{K'}^T X + b_{k'})} \tag{15}$$

Where $b_{k'}$s and $W_{k'}$s are the bias vector and weight, assuming k classes. The error function is denoted by (16).

$$J(W) = -\sum_{i=1}^{c} y_i \ln a_i \tag{16}$$

Where y_i is the actual label of the *i*-th sample and a_i is the predicted label for that sample.

Experimental Setup

The proposed hybrid model consisting of CNN and BiLSTM is applied in categorization of sentiment. It has an embedding layer, one convolution layer followed by two bidirectional LSTM layers and finally a fully connected dense classification layer. In this section, the datasets worked with, model hyperparameters settings, optimization techniques used, experiment results, evaluation results and model analysis are discussed in detail.

1) Dataset

The proposed model has been validated on the following two landmark sentiment datasets.

a) Internet Movie Database (IMDB) Review Dataset

The Stanford movie large review contains one lakh movie reviews from IMDB. It is segregated into 50,000 unlabeled training instances, 25,000 labeled training instances and remaining labeled testing instances. The training and test sets consist of disassociated set of movies, but balanced instances of positive and negative labels. It is worth noting that for any particular movie, a maximum of 30 reviews are allowed. This brings down the possibility of review correlation.

b) Stanford Sentiment Treebank2 Dataset (SST2)

It is a binary dataset having 11,855 Rotten Tomatoes reviews and a vocabulary size of around 16k. It is segregated into train, test and validation sets in the ratio of 70:20:10 respectively. It is mention worthy that this dataset can be trained at both sentence-level as well as phrase-level.

2) Model Settings

A visualization of the hybrid model is in Fig. 2.

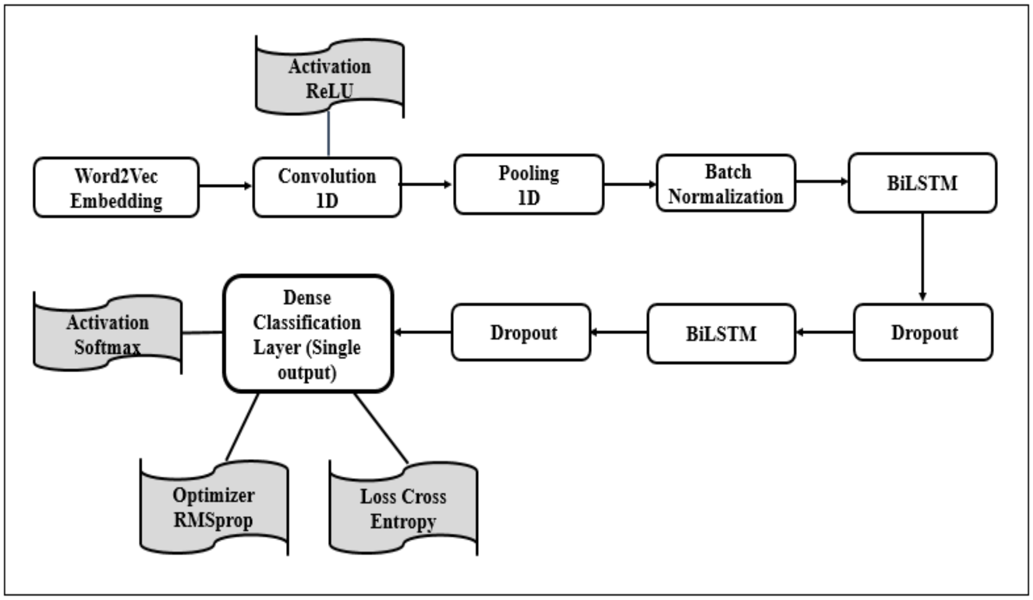

Fig. 2 Diagram of the proposed hybrid model

The Stanford IMDB sentiment analysis dataset has been preprocessed to a vocabulary size of 5000 words with max-min padding of 500 words for each user review, making any other data meaningless to the prediction outcome. As discussed earlier, for any particular movie, a maximum of 30 reviews are allowed.

In deep learning models, the neural network parameters and optimizations applied have significant impact on the performance of the classifier. Additional time has been devoted in model design aspects like initial weight, number of hidden layers, activation and loss functions, optimizers used etc. and in tuning hyperparameters like number of filters in CNN, learning and dropout rate among others.

Since LSTM layers can efficiently keep track of long-term dependencies and do not suffer from vanishing or exploding gradient problems, it was found out that adding more than one LSTM layers after a single one dimensional convolution layer optimally serves the purpose of our model.

In Conv1D layer the number of filters have been fixed as 128 each of variable width (4, 5, 5) and the number of generated feature maps have been fixed at 256. In order to decrease the input height by half, the kernel size has been set as 2. Rectified linear unit (ReLU) is used as the activation function and the receptive field size i.e. number of pixels of the input image affecting an element in a feature map is set between three to five. To optimally get rid of overfitting, the l2 kernel regularization parameter has been added with the convolution layer and it has been fixed to 0.01. Another efficient procedure of regularization is Dropout. Here, a dropout of 0.5 has been applied after the bidirectional LSTM layers. The number of units in the two BiLSTM layers has been set to 128 and 64 respectively.

The final fully connected classification layer has no parameters to tune. RMSprop has been employed as the optimizer as it yielded better result than ADAM. The weight decay and learning rate parameters were fixed at 0.01 and 0.1 respectively. Gradient clipping was also used to clip the gradients and cell outputs.

The proposed hybrid model has been trained for 100 epochs using Google Colaboratory GPU and CUDA v10.2 with batch sizes of 50 and 64 for the IMDB and SST2 datasets respectively. The model has been trained through stochastic gradient descent and cross entropy is used as the loss function. The train set is split into training and validation, keeping the latter's size same as that of the corresponding test set.

Evaluation Results and Analysis

The proposed model has been implemented using TensorFlow 2.0 on Keras. Training the model on both CPU and GPU, it is observed that training time on CPU takes eight to ten times more than that on GPU. The model achieves a training accuracy of 96% on the IMDB dataset and 91% on the SST2 dataset. Test accuracy and F1 score values are used to evaluate the performance of the model.

The accuracy achieved in this hybrid model on the IMDB dataset is 94.1%, while on the SST2 dataset an accuracy of 88.3% has been achieved. The close ranges of training and validation accuracy values confirm that the model does not suffer from overfitting. Accuracy is not a sufficient evaluation measure, in case of an uneven class distribution, which may be the case in many other datasets. This proposed model has a F1 score of .875 indicating high precision and recall.

A comparative analysis of the proposed model with that of some other prominent existing methods, on similar IMDB dataset is shown in TABLE 1.

Table 1 Model Accuracy on IMDB Dataset

Method used in Model	**Accuracy (%)**
SA-LSTM with joint training [28]	85.3
CNN	85.6
LSTM with tuning and dropout [28]	86.2
CNN-LSTM [29]	90.7
Proposed Hybrid model, CNN + Bi-LSTM	94.1

This work also attempted sentiment classification on the SST2 dataset using only CNN, LSTM and an ensemble of CNN-LSTM. A comparative study with the proposed hybrid model is presented in TABLE 2.

Table 2 Model Accuracy on SST2 Dataset

Method used in Model	**Accuracy (%)**
CNN	81.6
LSTM	83.2
CNN-LSTM	85.1
Proposed Hybrid model, CNN + BiLSTM	88.3

From these results, it is obvious that the model achieves good results in mining deep sentiment representation for emotion categorization due to the ensemble of CNN and BiLSTM.

Conclusion

This paper shows the efficacy of the proposed model in sentiment classification on two landmark sentiment datasets. Word2Vec word embeddings are used in representing each review words and the embeddings are then fed into the stack of CNN and BiLSTM layers. The convolution and pooling layers of CNN takes care of local features while the LSTM layers facilitate in learning sequence characteristics. As future enhancements, other popular word embeddings like Wiki FastText embeddings may be tried to improve pre-training of review words as the input to the deep network. Other deep learning models like Gated Recurrent Units (GRU) and attention mechanisms may be merged with this model to further improve the performance.

Acknowledgements

The authors are appreciative to the Presidency University, Bengaluru for presenting all the foundations for the direct of this exploration. They want to thank the editors and all the anonymous critics for their valuable remarks and proposals in enhancing the nature of this paper.

References

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* 655–665.

Mikolov, T., Karafiát, M., Burget, L., & Cernocky, J. (2010). Recurrent neural network based language model. *In Interspeech*, *2*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. *In Proceedings of the conference on empirical methods in natural language processing*, 1422-1432.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097-1105.

Sainath, T.N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.R., Dahl, G., & Ramabhadran, B. (2015). Deep convolutional neural networks for large-scale speech tasks. *Neural networks*, *64*, 39-48.

Pang, B., Lillian, L., & Shivakumar, V. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics.*

Liu, H., Lieberman, H., & Selker, T. (2003). A model of textual affect sensing using real-world knowledge. *In Proceedings of the 8th international conference on Intelligent user interfaces*, 125-132.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*, 1137-1155.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111-3119.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational linguistics*, 1746-1751.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *In Advances in neural information processing systems*, 649-657.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Socher, R., Huval, B., Manning, C.D., & Ng, A.Y. (2012). Semantic compositionality through recursive matrix-vector spaces. *In Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning*, 1201-1211.

Tai, K.S., Socher, R., & Manning, C.D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual*

Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1556–1566.

You, Q. (2015). Robust image sentiment analysis using progressively trained and domain transferred deep networks. *Twenty-Ninth AAAI Conference on Artificial Intelligence.*

Lakkaraju, H., Socher, R., & Manning, C. (2014). Aspect specific sentiment analysis using hierarchical deep learning. *In NIPS Workshop on deep learning and representation learning*, 1-9.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.* A field guide to dynamical recurrent neural networks. IEEE Press, 1-15.

Xiao, Y., & Cho, K. (2016). Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint arXiv:1602.00367*.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, *18*(5-6), 602-610.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*, 2493-2537.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107-116.

Dai, A.M., & Le, Q.V. (2015). Semi-supervised sequence learning. *In Advances in neural information processing systems*, 3079-3087.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J.F., & Iglesias, C.A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, *77*, 236-246.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1253.