A Survey of Deep Learning for Sentiment Analysis

Abhinandan P Shirahatti
Department of Computer Science and Enggineering
KLE Dr. M. S. Sheshgiri College of Engineering and Technology,
Belagavi, India
abhinandanshirahatti@klescet.ac.in

Krupa R Rasane
Department of ECE
Jain College of Engineering
Belagavi, India
hod_ec_krr@jainbgm.in

***Abstract*—Deep learning has detonated in the public responsiveness, primarily as predictive and analytical products pervade our world, in the form of innumerable human-centered smart-world systems, including targeted advertisements, natural language assistants and interpreters, and mock-up self-driving vehicle systems. In contrast, researchers across disciplines have been including into their research to solve various natural language processing issues. In this paper we seek to provide a thorough exploration of Deep learning and its applications like sentimental analysis and natural language processing (NLP). Deep learning has an edge over the traditional machine learning algorithms, like support vector machine (SVM) and Naïve Bayes, for sentiment analysis because of its potential to overcome the challenges faced by sentiment analysis and handle the diversities involved, without the expensive demand for manual feature engineering. Deep learning models promise one thing - given sufficient amount of data and sufficient amount of training time, they can perform the task of sentiment classification on any text class with minimal restrictions and no task-specific or data-specific manual feature engineering. We hope this survey provides a valuable reference for new Deep learning practitioners, as well as those seeking to innovate in the application of deep learning.**

Keywords— Deep learning, Support vector machine (SVM), Naïve Bayes, Sentiment analysis, Natural language processing (NLP)

I. INTRODUCTION

Deep learning has come to influence industry and research spheres for the development of a variety of smart-world systems, and for good cause. Deep learning has shown significant potential in approximating and reducing large, complex datasets into highly accurate predictive and transformational output. Previous research work has shown that basic machine learning techniques produce effective results in performing several natural language processing tasks like topic categorization of documents. However the same techniques cannot be naively used for sentiment classification. The non-trivial nature of the latter demands extra effort to contribute effectively towards opinion classification. Opinions need more understanding for them to be analysed properly. We discuss some techniques from the two machine learning paradigms: traditional models, which have proved useful for sentiment analysis since over the past few decades, and deep learning models, which have emerged as a powerful tool for natural language processing in recent years.

II. TRADITIONAL PARADIGM

Substantial research has been done over the past few years to exploit popular machine learning algorithms for the task of sentiment classification [1]. Depending on the problem statement in sentiment mining, these classifiers Have shown good performance accuracy, provided proper feature engineering and pre-processing steps are carried out prior to the classification process.

A. Conventional Models

Naïve Bayes Classifier is the simplest and the most widely used probabilistic classification algorithm [2]. It is based on Bayes' Theorem. It basically calculates the posterior probabilities of events and assigns the label with the maximum posterior probability to the event.

A major assumption made by the Naive Bayes Classifier is that the features are conditionally independent, given the sentiment class of the document [1], which is not true in real-life situations. Furthermore, another problem with this technique is that, if some feature value, which was not encountered in the training data, is seen in the input data, its corresponding probability will be set to 0. Bayes classifier fails in this case. To remove this undesirable effect, smoothing techniques are applied.

Maximum Entropy classifier is another model which performs probabilistic classification, making use of the exponential model. It is based on the Principle of Maximum Entropy which states that subject to the prior data which has been precisely stated, the probability distribution which describes this data with the current knowledge in the best possible manner is the one with the largest possible entropy value. This technique has been proven to be effective in many NLP classification tasks including sentiment analysis.

Max entropy classifier is seen to outperform the Naive Bayes in many cases [1]. One major advantage of this classifier is that it makes no conditional independence assumption on the features of the documents to be classified, given a sentiment class. Hence, it is applicable to real-life scenarios, unlike in case of Naive Bayes.

Support Vector Machines (SVMs) (Cortes and Vapnik, 1995) have proved to be highly effective for the categorization of documents based on similar topics. As opposed to the probabilistic classifiers like the previous two [2], this method aims to find large margin between the different classes. It is a supervised learning model which analyses data and learns patterns which can be used to classify the data.

Support vector machines attempt to find a hyper plane (in case of 2-class classification problem) which not only separates data points based on the category they belong to, but also tries to maximize this separation gap between the two classes, i.e., this is a constrained optimization problem. One major advantage of this classifier is that it makes no assumption on the documents to be classified and it endeavours to find the best classification margin for the data at hand instead of relying on probability values. It is one of the widely used machine learning algorithms, which yields very good results for the task of sentiment analysis [1].

B. *Possible Limitations*

Most of the classical machine learning algorithms for text classification are either rule-based or corpus-based. Their efficiency depends on the quality of the annotated corpora as well as the feature engineering task involved prior to the classification. The features need to be manually handcrafted as well as they differ from domain to domain and document to document, which makes it less generic and more text-specific. The accuracy of these systems depends on how the features were chosen, which makes the system liable. Furthermore, it is very difficult, and many a times not feasible, to adapt a system designed for a particular problem to new problems or in different language for the same problem. And for texts like tweets, which do not follow any rules or grammar as such, these approaches tend to perform very badly. Hence, extensive pre-processing and feature engineering need to be done specific to the text genre, language and the problem statement using other NLP tools since these tools are not 100% accurate, the loss in accuracy in the pre-processing steps will in turn affect the overall accuracy of the sentiment analysis task. Hence, pre-processing steps, especially feature extraction steps, need to be carefully managed. Although good accuracy values have been reported in literature and these algorithms seem to work well for a long time, there is a large scope of improvement, which cannot be overlooked.

III. INTRODUCTION TO DEEP LEARNING

Deep Learning is a subpart of machine learning concerned with algorithms energized by the structure and function of the brain called artificial neural networks.

Neural networks recently have become a very popular topic of research in the field for natural language processing(NLP), including sentiment analysis. Neural networks are proving useful in solving almost any machine learning classification problem. The only adjustment required is defining its architecture — number of hidden layers to be used, number of hidden units to be present in every layer, activation function for every node, error verge for the data, the type of inter-connections, etc.

Once a suitable neural network architecture is designed for the problem at hand, a solution to the classification problem can be obtained using deep learning models. The only demand for deep learning models is enough training data and enough time and resources to train the network for classification. Clearly, a traditional machine learning algorithm can be designed using deep learning but not necessarily vice-versa. This is because neural networks are capable of capturing very complex characteristics of data without any significant involvement of manual labour as opposed to the machine learning systems. Deep learning uses deep neural networks to learn good representations of the input data, which can then be used to perform specific tasks.

B. Basic Neaural Network

Neural Networks play an important role in machine learning and cognitive science. These have been widely used in the field of image processing and pattern recognition. Recently, they are becoming popular for solving Natural Language Processing problems. A neural network can be used to learn the word embeddings as well as in turn use them as input for NLP tasks like sentiment classification. The basic structure of a fully-connected neural network, which uses one hidden layer is shown in Figure 1.

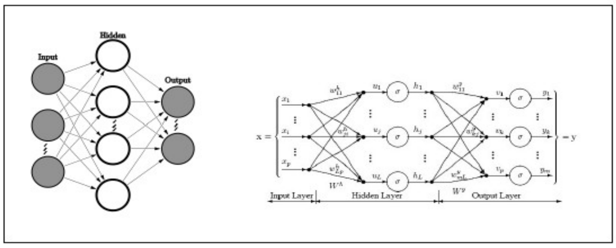

Figure 1: A Fully-Connected Simple Neural Network [4]

The weights on edges are learned by means of back-propagation of errors through the layers of the neural network based on the inter-connections and the non-linear functions. The labelled data is fed to the network a number of times (called epochs) for the network to learn the weight parameters until the error becomes negligible (in the ideal case). Generally, for experiments, training is done for a fixed number of times to reach a minimum error value when the network does not converge any further.

IV. WORD EMBEDDINGS

Neural networks in NLP, unlike other traditional algorithms, do not take raw words as input, since the networks can understand only numbers and functions. Hence words need to be transformed into feature vectors, or in other words word embedding [5], which capture the characteristics and semantics of the words if extracted properly. The word vectors can be learned by feeding large raw corpus into a network and training it for sufficient amount of time. presented the first large-scale deep learning model for natural language processing to learn the distributed representation of words by using language modelling (Figure 2. The network is trained on a raw corpus, which is expressed as sequence of words. The idea is (a) to associate each word in the vocabulary with a real-valued word feature vector of m dimensions, (b) to express the joint probability function of word sequences in terms of the feature vectors of the words occurring in the sequence, and (c) to learn the word feature vectors and the parameters of the probability function simultaneously.

Figure 2: Bengio's Neural Network Model[4], Each word embedding may be of any dimensionality as the user wishes. Higher dimensionality implies more information captured but on the other hand incurs higher computational expense.

Hence, a trade-off is to be chosen to balance both. Google has released pre-trained vectors trained on part of Google News dataset (about 100 billion words)1, which can be used by researchers.

The model contains 300-dimensional vectors for 3 million words and phrases. These can then be used as inputs into the neural networks for any NLP tasks. The quality of the word vectors is defined by how the vectors distinguish between dissimilar words and are close for similar ones. The closeness of word vectors is generally determined by cosine distance [5].

V. NEURAL NETWORK ARCHITECTURES FOR NLP

Generally, for NLP tasks, we tend to use the Window Approach [6]. This method assumes that the tag to be assigned to a word in a sentence depends upon its neighbouring words. Hence, a fixed window size (additional hyper-parameter) is chosen and this amount of words is fed into the network to tag the middle word (Figure. 3). The feature window is not defined for border words (start/end) and hence padding is done in the sentences.

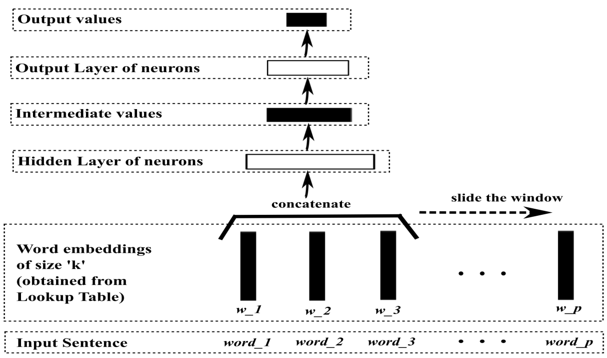

Figure 3: Window-level Neural Network Architecture [6]

The window-level approach cannot be applied to macro-text level tasks like sentiment classification because sentiment tag requires the whole sentence to be taken into consideration, whereas in window level approach, only a portion of sentence is considered at a time. Also, for other NLP tasks as well, one word may depend on some word which does not fall in the pre-decided window. Hence, sentence-level approach is a viable alternative, which takes the feature vectors of all the words in the input text as input [6]. Figure 4 shows the overall structure of such a network which takes the whole sentence as input. The sequence layer can have different structures to handle the sequence of words in the text. For sentence classification, since sentences can be of variable size, there is a pooling layer after the sequence layer, which is a feature map of a fixed size.

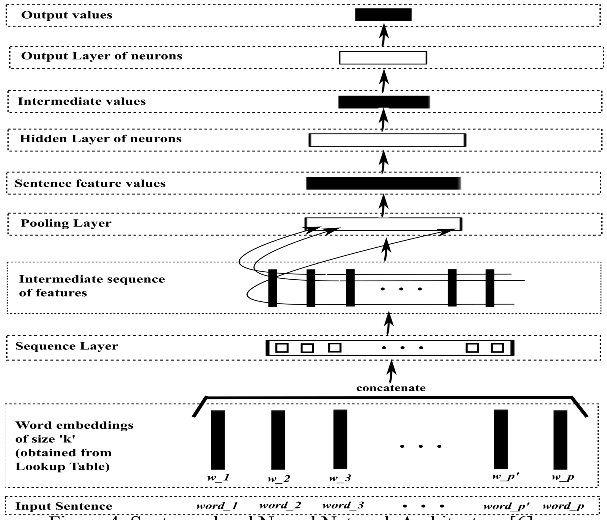

Figure 4: Sentence-level Neural Network Architecture [6]

The number of layers, the number of units in each layer, the structure of the sequence layer, the dimensionality of the word vectors, the interconnections and the activation functions are some of the hyper-parameters of the neural network model, which need to be tuned to achieve the best performance for a particular task.

a. Convolutional Neural Networks (CNN)

Proposed a unified neural network architecture [6] which can be applied to numerous Natural Language Processing tasks like Part-Of-Speech Tagging, Parsing, Chunking, Semantic Role labelling and Named Entity Recognition. The architecture, known as CNN (Convolutional Neural Network), takes concatenated word vectors of the text as input and involves convolutional and max-pooling layers prior to the general neural network framework.

Convolution Layer: It is a sort of generalization of window approach where a window of fixed size is moved over the sentence and the weight matrix is same for each sequence. One feature vector is obtained by convoluting over each sequence. This layer is meant to extract local features from sequence of words in the sentence. The network can have a number of window sizes and a number of weight matrices, each forming one channel.

Max-Pooling Layer: The length of the output of convolution layer depends on length of the input sentence and the number of channels used. To establish uniformity in the size of the sentence vector, max-pooling layer is used to select the maximum value for each feature across all windows. This is preferred over simple averaging because for the classification, all words do not contribute equally; their relative significance is captured by the max-pooling layer. Now the global feature vector size for the sentence is proportional to that of individual words and to the number of channels used, i.e., its length is constant for all sentences of varying length.

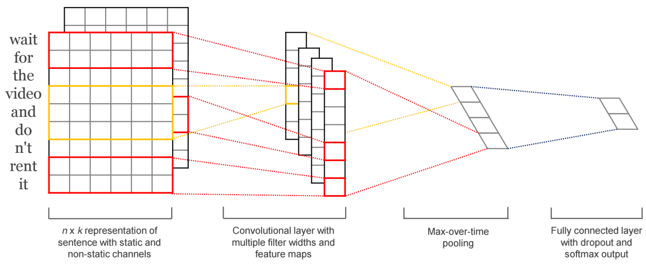

Figure 5: Model architecture with two channels in CNN [7]

The "sentence vector" is then fed into a fully connected neural network with 0/1/2/.. Hidden layers and activation functions like softmax or sigmoid to ultimately reach the output layer whose size is equal to the number of labels. A model architecture of CNN is shown in Figure 5 [7].

b. *Recursive Neural Tensor Networks (RNTN)*

A recursive neural tensor network (RNTN) [8] is a kind of deep learning model in which the same set of weights is applied recursively over a structure (e.g. tree), to produce a structured or a scalar prediction over variable length input, by traversing the given structure in topological order. The RNTN model takes as input, the word vectors and the parse tree of the text, and then computes vectors for the nodes in the tree using a single tensor-based composition function. This model is a modification over the recursive neural networks which uses a simple weight matrix shared across all the nodes. The input sentence is first converted into a constituent parse tree. The leaves of the tree are represented by the corresponding word vectors.

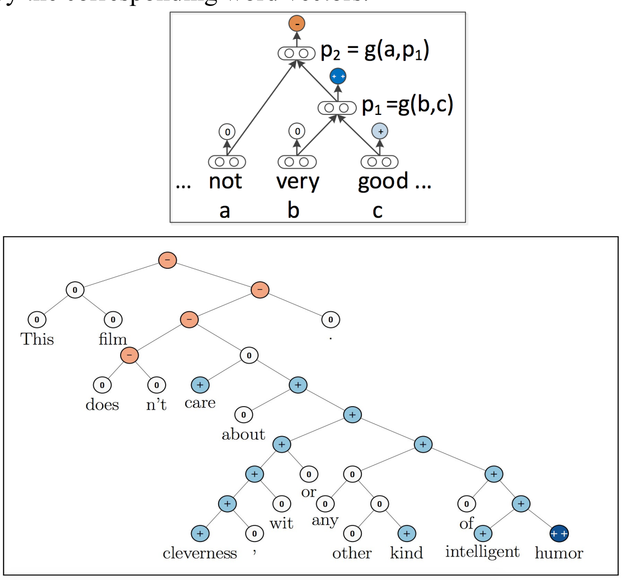

Figure 6: Example of Recursive Neural Tensor Network [8].

c. *Recurrent Neural Networks (RNN)*

Recurrent Neural Network Models [5] are a form of neural networks which do not depend on the window size to work for Natural Language Processing tasks. RNN is capable of conditioning the network on all the previously seen inputs (words in case of a sentence). In addition to dependency on the current input, the value of each hidden layer unit also depends on its previous state, thereby propagating the effects of words over the sentence (Figure 7).

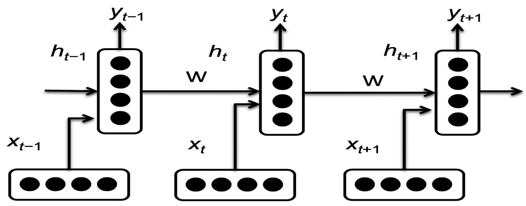

Figure 7: A Recurrent Neural Network with three time steps [5]

The word vectors for each word is fed into the network one by one and the effect of each word is carried on till the end of the sentence, thereby ensuring that the dependency of each word on all other words is captured through activations of neurons and back-propagation on weight matrices. The goal of an RNN implementation is to allow propagation of context information through faraway time-steps. RNN model works by propagating weight matrices over the time-steps. However, this creates anomalies which are not acceptable in practice. Intuitively, one should be able to predict a word more accurately given more context (i.e., more number of words preceding this) as compared to lesser context. However, RNN tends to perform the opposite due to the problems of Vanishing Gradient and Gradient Explosion problems.

d. *Long Short Term Memory (LSTM)*

Long Short Term Memory networks are a modified version of the recurrent neural networks but with much more complicated activation units. The key element of an LSTM model is a memory cell. Here, information is stored in two ways (hence the name Long Short Term Memory):
Short-term Memory as activations of the neurons which capture the recent history, Long-term Memory as weights which are modified based on back propagation This model allows retention of information over a much longer period (more than the usual 10-12 steps as in case of RNNs) through the use of the memory cell and hence produces appreciable results when applied to NLP tasks. The internal units of an LSTM model are shown in Figure 8. The network architecture is very complex and its structure can be broken down into certain stages:

1. Input Gate: It uses the input word and the past hidden state to determine whether or not the current input is worth preserving.

2. New Memory Generation: It uses the input word and the past hidden state to generate a new memory which includes aspects of the new word.

3. Forget Gate: It uses the input word and the past hidden state to make an assessment on whether the past memory cell is useful for computation of the current memory cell.

4. Final Memory Generation: It first takes the advice of the forget gate and accordingly forgets the past memory; it then takes the advice of the input gate and accordingly gates the new memory and lastly it sums these two results to produce the final memory.

5. Output/Exposure Gate: It makes the assessment regarding what parts of the memory needs to be exposed/present in the hidden state.

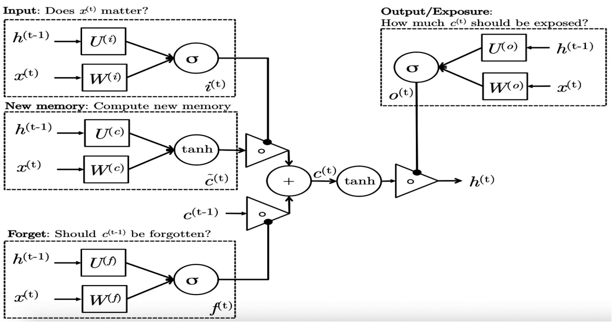

Figure 8: Detailed Internals of an LSTM

The task of sentiment analysis is generally assigning a label to the whole sentence and not to the individual words.

Hence, the outputs produced at each time-step, i.e., with each input word, need to be reconciled to finally get a single label for whole sentence. One way is to have a number of LSTM cells for each of which the outputs are passed through a mean-pooling layer before going through logistic regression. Hence, in spite of the fact that the LSTM network is very complicated and has its own disadvantages like huge computational complexity, the network offers promising results since it is capable of taking into account the whole sentence as context to generate results.

VI. Performance of Models

TABLE I. ANALYSIS OF CONVOLUTIONAL NEURAL NETWORKS

Researcher Name and Year	Model Used	Purpose	Data Set	Results
J.Islam and Y. Zhang 2016 [14]	ConvolutionalNeural Networks (CNN)	Visual SA	1269 images from twitter	GoogleNet gave almost 9 % performance progress than AlexNet.
A. Severyn and A. Moschitti, 2015 [15]	CNN	Phrase level and message level task SA	Semeval-2015	Compared with official system ranked 1st in terms of phrase level subtask and ranked 2nd in terms of message level.
Q. You, J. Luo, H. Jin, and J. Yang, 2015 [16]	Convolutional Neural Networks (CNN)	Textual-visual SA	Getty Images, 101 keywords	Joint visual and textual model outperforms the early single fusions.
X. Ouyang, P. Zhou, C. H. Li, and L. Liu, 2015[17]	Convolutional Neural Networks (CNN)	Sentiments of sentences	rottentomatoes.com (contains movie review excerpts)	The proposed model outperformed the previous Models with the 45.5% accuracy.

TABLE II. ANALYSIS OF RECURSIVE NEURAL NETWORKS

Researcher Name and Year	Model Used	Purpose	Data Set	Results
C. Li, B. Xu, G. Wu, S. He, G. Tian, and H. Hao, 2014 [18]	Recursive Neural Deep Model (RNDM)	Chines sentiments analysis of social data	2270 movie reviews from websites	Performs higher (90.8%) than baselines with a great margin..
R. Socher, A. Perelygin, and J. Wu, 2013 [19]	RNTN (Recursive Neural Tensor Network)	Semantic compositionality	11,855 single sentences from movie review (Pang and Lee2005)	The RNTN achieved 80.7% accuracy in sentiment prediction, an improvement of 9.7 % over baselines (bag of features).
W. Li and H. Chen, 2014 [20]	Recursive Neural Network (RNN)	Identifying Top Sellers In Underground Economy	Russian carding Forum)	Results have been indicated that Deep learning techniques accomplish superior outcomes than shallow classifiers. .
A. Hassan, M. R. Amin, A. Kalam, A. Azad, and N. Mohammed, [21]	Deep Recurrent model especially LSTM (Long Short Term Memory	Sentiment Analysis on Bangla and Romanized Bangla Text (BRBT)	9337 post Samples from different social sources	Ambiguous Removed with 78% accuracy. Ambiguous converted to 2 scored highest with 55% accuracy.

T. Chen, R. Xu, Y. He, Y. Xia, and X. Wang , 2016 [21]	Recurrent Neural Network (RNNGRU)	Learning User and Product Distributed Representation	Three datasets collected from Yelp and IMDB.	Results have been indicated that proposed model outperformed many baselines including RNN

Table 3. The results of Precision, Recall and F1-measureusing GloVe-DCNN and baseline. BoW refer to uni- and bi-gram features. GloVe refer to concatenate BoW vectors with the average GloVe representations, word sentiment polarity feature and twitter-specific feature. DCNN refers to deep convolution neural network. BoW-SVM represents the use of the SVM classifier and the BoW features vector [23].

Method	Positive /%			Negative /%			Average /%		
SED	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BoW-SVM	66.40	96.39	78.52	83.86	27.04	40.54	75.13	61.71	59.53
BoW-LR	86.92	83.05	84.89	76.11	81.45	78.60	81.51	82.25	81.74
GloVe -SVM	86.82	90.22	88.42	84.30	79.61	81.74	85.56	84.92	85.08
GloVe -LR	86.81	89.42	88.96	83.72	83.89	83.20	86.16	86.15	86.08
GloVe-DCNN	87.29	92.39	89.77	88.71	82.25	85.35	**88**	**87.32**	**87.66**
SSTd	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BoW-SVM	64.86	88.37	74.77	66.20	32.43	43.50	65.53	60.40	59.13
BoW-LR	77.49	74.79	75.58	66.59	68.48	66.34	72.04	71.64	70.96
GloVe -SVM	78.89	87.88	83.07	79.53	67.25	72.73	79.21	**77.57**	77.90
GloVe -LR	79.41	82.27	80.72	73.52	70.22	71.62	76.46	76.24	76.18
GloVe-DCNN	85.03	83.46	84.23	76.19	78.26	77.21	**80.61**	**80.86**	**80.72**
STSGd	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BoW-SVM	80.00	11.88	8.04	69.35	99.64	80.06	74.68	55.76	44.05
BoW-LR	67.38	55.59	47.25	77.34	93.70	83.42	72.36	74.64	65.34
GloVe -SVM	70.74	61.59	53.21	79.43	94.57	85.21	75.09	78.08	69.21
GloVe -LR	63.76	70.00	56.60	82.05	89.21	84.71	72.91	79.61	70.66
GloVe-DCNN	75.35	74.85	75.06	90.15	90.34	90.24	**82.75**	**82.61**	**82.65**
SE2014	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BoW-SVM	74.65	97.55	84.58	61.64	10.61	18.11	68.15	54.08	51.34
BoW-LR	79.26	91.87	85.10	61.57	35.14	44.74	70.42	63.51	64.92
GloVe -SVM	86.68	91.85	89.19	73.80	61.94	67.35	80.24	76.89	78.27
GloVe -LR	86.74	88.26	87.49	66.75	63.59	65.13	76.74	75.92	76.31
GloVe-DCNN	75.93	71.93	73.87	89.19	91.03	90.10	**83.56**	**81.48**	**81.99**
STSTd	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BoW-SVM	76.81	59.33	65.93	65.32	81.86	71.75	71.07	70.60	68.84
BoW-LR	75.91	65.35	69.39	68.42	79.14	72.64	72.17	72.25	71.01
GloVe -SVM	82.18	83.30	81.70	81.62	82.20	80.93	81.90	82.75	81.32
GloVe -LR	82.10	82.68	81.60	81.08	82.60	81.06	81.59	82.64	81.33
GloVe-DCNN	87.98	89.47	88.71	87.22	85.42	86.29	**87.60**	**87.45**	**87.50**

VI. Conclusion

Employing deep learning to sentiment analysis has become a prominent research topic lately. In this paper, we introduced many deep learning architectures and their applications in sentiment analysis. Many of these deep learning techniques have shown state-of-the-art results for various sentiment analysis tasks. With the advances of deep learning research and applications, we believe that there will be more exciting research of deep learning for sentiment analysis in the near future.

References

[1] Thumbs up? Sentiment Classification using Machine Learning Techniques. (EMNLP), Philadelphia, July 2002, pp. 79-86. Proceedings of the Conference on Empirical Methods in NLP.

[2] Jason D. M. Rennie,Tackling the Poor Assumptions of Naive Bayes

[3] Text Classifiers, Artificial Intelligence Laboratory; Massachusetts Institute of Technology; Cambridge, MA 02139,2003

[4] Andrea Esuli∗ and Fabrizio Sebastiani, SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining,2005-06.

[5] Yoshua Bengio, A Neural Probabilistic Language Model, Journal of Machine Learning Research 3 (2003) 1137–1155, 2003.

[6] Tomas Mikolov Google Inc. Efficient Estimation of Word Representations in Vector Space, Machine Learning Research Sep 2013.

[7] Ronan Collobert, Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research 12 (2011) 2493-2537.

[8] Yoon Kim, Convolutional Neural Networks for Sentence Classification Journal of Machine Learning Research 3 Sep 2014.

[9] Richard Socher, Alex Perelygin, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Google scholar,2013.

[10] Felix A. Gers, Learning Precise Timing with LSTM Recurrent Networks, Journal of Machine Learning Research 3 (2002) 115-143.

[11] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. Computational linguistics, 22(1):39–71.

[12] Roberto González-Ibáñez, Identifying Sarcasm in Twitter: A Closer Look, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 581–586, Portland, Oregon, June 19-24, 2011.

[13] P. Ruangkanokmas, T. Achalakul, and K. Akkarajitsakul, Deep Belief Networks with Feature Selection for Sentiment Classification, Uksim.Info, pp. 16, 2016.

[14] C. Baecchi, T. Uricchio, M. Bertini, and A. Del Bimbo, A multimodalfeature learning approach for sentiment analysis of social network multimedia,Multimed. Tools Appl., vol. 75, no. 5, pp. 25072525, 2016.

[15] J. Islam and Y. Zhang, Visual Sentiment Analysis for Social ImagesUsing Transfer Learning Approach, 2016 IEEE Int pp. 124130, 2016.

[16] A. Severyn and A. Moschitti, Twitter Sentiment Analysis with Deep Convolutional Neural Networks, Proc. 38th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR 15, pp. 959962, 2015.

[17] Q. You, J. Luo, H. Jin, and J. Yang, Joint Visual-Textual Sentiment Analysis with Deep Neural Networks, Acm Mm, pp. 10711074, 2015.

[18] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, Sentiment Analysis Using Convolutional Neural Network, Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM), 2015 IEEE Int. Conf., pp. 23592364, 2015.

[19] C. Li, B. Xu, G. Wu, S. He, G. Tian, and H. Hao, Recursive deep learning for sentiment analysis over social data, Proc. - 2014 IEEE/WIC/ACM Int. Jt. Conf. Web Intell. Intell. Agent Technol. - Work. WI IAT 2014, vol. 2, pp. 13881429, 2014.

[20] R. Socher, A. Perelygin, and J. Wu, Recursive deep models for semantic compositionality over a sentiment treebank, Proc. , pp. 16311642, 2013.

[21] W. Li and H. Chen, Identifying top sellers in underground economy using deep learning-based sentiment analysis, Proc. - 2014 IEEE Jt. Intell. Secur. Informatics Conf. JISIC 2014, pp. 6467, 2014.

[22] A. Hassan, M. R. Amin, A. Kalam, A. Azad, and N. Mohammed,Bangla Text (BRBT) using Deep Recurrent models.

[23] ZHAO JIANQIANG, "Deep Convolution Neural Networks for Twitter Sentiment Analysis", VOLUME 6, 2018 2169-3536 2018 IEEE. Translations.