International Journal on Emerging Technologies **10**(4): 38-42(2019)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Text Classification using LSTM based Deep Neural Network Architecture

***Sheelesh Kumar Sharma*[1] *and Navel Kishor Sharma*[2]**
[1]Professor, Department of MCA, IMS Ghaziabad, (Uttar Pradesh), India.
[2]Associate Dean, Academic City College, Ghana.

(Corresponding author: Sheelesh Kumar Sharma)

(Received 12 July 2019, Revised 20 September 2019 Accepted 07 October 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Text Classification is important due to the need of managing the large amount of data by dividing them into known categories. Automatic classification of voluminous data in desired categories is still a challenge. Traditional approaches fail while dealing with huge amount of data and tend to degenerate with variety. Deep learning is helpful in text classification and can learn from voluminous data. Deep learning based methods can learn from voluminous data and can yield better results in general pattern recognition tasks. The paper presents a deep learning approach using LSTM, which can learn from data. LSTM has the obvious advantage over RNN as it overcomes the exploding and vanishing gradient problems. The experimental results on benchmarked data sets show that method is accurate. The approach has been tested on three different data sets of text classification and has been compared with two other methods. The average accuracy of the proposed technique is 80.90%. Experimental results show that proposed method can work with multi-class as well as binary text data and is better than other methods.

Keywords: Text classification, deep learning, LSTM, sentiment analysis, RNN, CNN.

I. INTRODUCTION

Text classification is important due to its applications in a variety of domains. The conventional techniques such as SVM, ANN, Decision Forests etc have the limitation of scalability of large data. There are following applications of text classification:

(i) Sentiment Analysis on Social Media: Using text classification techniques, the sentiments of people can be tagged as supportive, against, neutral etc on a given topic of discussion. Sentiment analysis is also termed as opinion mining and is a great tool for estimating the polarity of views expressed.

(ii) News Categorization: A given news can be classified in already known categories such as politics, entertainment, sports, crime etc. It helps the news portals and users to get a more relevant grouping of news headlines.

(iii) Spam Filtering: Text categorization techniques also help achieve the goal of tagging a piece of text (generally an email or text message) as spam or ham.

(iv) Customer Feedback/Complaint Grouping: Nowadays, almost all companies have a platform to connect to their customers to know the feedback of their products and services. Manual processing of customer feedback is not possible. Automatic filtering of feedback and complaints using text classification strategies saves the companies from lots of pain and improves their efficiency in fast redressal and closure of complaints.

(v) Movie and Product Reviews: This is another emerging area of text classification, where reviews received from thousands of people on products and movies can be classified into categories such as positive, negative or neutral. It helps in making many business decisions.

(vi) Document Classification: Right classification of a large number of documents into right categories helps in their easy management. It can be achieved based on document content analysis. This application of text classification helps find the solution of many problems like research paper classification based on the content, web page classification etc.

All the above listed applications have great commercial value. It makes the field of text classification more prominent and an active area of research. Available approaches for text classification are not able to solve the problems encountered in these applications.

Many of the text classification approaches use bag-of - words approach. In this approach a vocabulary is created from all available sentences and then represented in the form of a feature vector. For example consider the following two sentences:

S1 = {“Ravi loves bananas”}
S2 = {“Mohan loves mangoes”}

The vocabulary is the set of all unique words indexed in an order. Out of above two sentences S1 and S2 we get the vocabulary of 5 distinct words as follows:

Voc={‘Mohan’:0, ‘Ravi’:1, ’bananas’:2, ’loves’:3, ’mangoes’:4}

Above two sentences can be represented as feature vectors using vocabulary Voc as a 2-D array {[0, 1, 1, 1, 0], [1, 0, 0, 1, 1]}. For a given sentence structure, the places in array have been filled with 1 wherever it matches with words in sentence and others with 0.

II. RELATED WORK

Both CNN and RNN have been used for sentence modeling and text classification. They are the better approaches as compared with others as it is easy to capture spatial and temporal features in text. With the ability to capture spatio-temporal features CNN based methods have achieved competitively high accuracy in general tasks of natural language processing, speech recognition and computer vision. The text classification can be a binary or multi-class classification problem. Multi-class classification problem can be solved in a variety of ways including as a transformation or extension from binary classification.

Multi-class classification can be solved as a hierarchical problem.

The transformation from binary approaches follow either one vs one strategy or one vs all strategy of classification. Various binary classifiers can adapt to solve the problem of multi-class classification as well. The popular methods include decision trees, k-nearest neighbors, naive Bayes, support vector machines, artificial neural networks.

Based on the length of the text used for classification, the task can be divided at three levels namely short text classification, medium text classification, and large text classification problem. The short text classification includes small pieces of text such as tweets, which a system is trained to perform classification on. Middle level text classification involves the text fragments ranging from 500 words to 1500 words and includes the blog posts, web pages etc. The large text classification involves the fragments of texts generally having many thousands of words such as books and other long documents. CNN based text classification systems have an edge over word embedding based techniques.

Adamuthe *et al.,* [1] provide a comparative study of CNN with word embedding approaches for text classification. A multi class text classification approach has been presented in [2] using online topic models. In a variety of applications, text classification can be associated with other tasks such as speech recognition. In the work of Akhtiamov *et al.,* [3], speech recognition and text classification have been used to solve the problem of addressee classification. Their work presents a useful comparison of deep classifiers with conventional textual classifiers.

Mining the social media data can be fruitful in many ways. Yada *et al.,* [4] present a mechanism for identifying the tweets having the mention of a book. They eliminate the possibility of potential spam and promotional tweets posted by automated bots and take into account only the real ones. There exist the multi-modal approaches for text classification, wherein the knowledge from various forms of text can be combined to achieve the objectives of the task.

Melville *et al.,* [5] proposed such a technique that harnesses the lexical knowledge for text classification. String kernels turn to be useful for the task and they have been used for text classification. In [6], a string kernel is used for accomplishing the task of text classification. Here, kernel is calculated as the inner product of sub-sequences of finite length. Similar to bag of words, some researchers have built on similar concepts.

Joulin *et al.,* [7] describe a text classification strategy called bag of tricks.

This approach is faster and more efficient than other approaches of text classification as discussed in their work [7].

III. DEEP LEARNING MODELS AND ARCHITECTURES TEXT CLASSIFICATION

Deep learning models are the extension of the concept of Artificial Neural Networks (ANN) having evolution to Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) networks, bidirectional networks, temporal networks etc. There are many popular architectures based on these networks which are used in a finding solutions to various tasks such as computer vision, speech recognition and also to natural language processing. These architectures include Resnet, VGG 16, VGG 19 etc.

IV. WHAT IS LSTM?

Long Short Term Memory network is a type of Recurrent Neural Network (RNN). A typical LSTM has four components namely- cell, input unit, output unit and forget unit. The latter three units are also called as gates. Cell is the memory part of LSTM, which keeps track of the relationships among data variables. Input controls flow of data entering the network, forget unit controls how long to keep the data in memory, and output units produces desired output.

LSTM can be recurring with multiple hidden layers. Multi layered LSTM architecture can yield better results than single layer LSTM in some applications like speech recognition, and natural language processing. Fig 1 shows the basic architecture of a recurrent LSTM network.

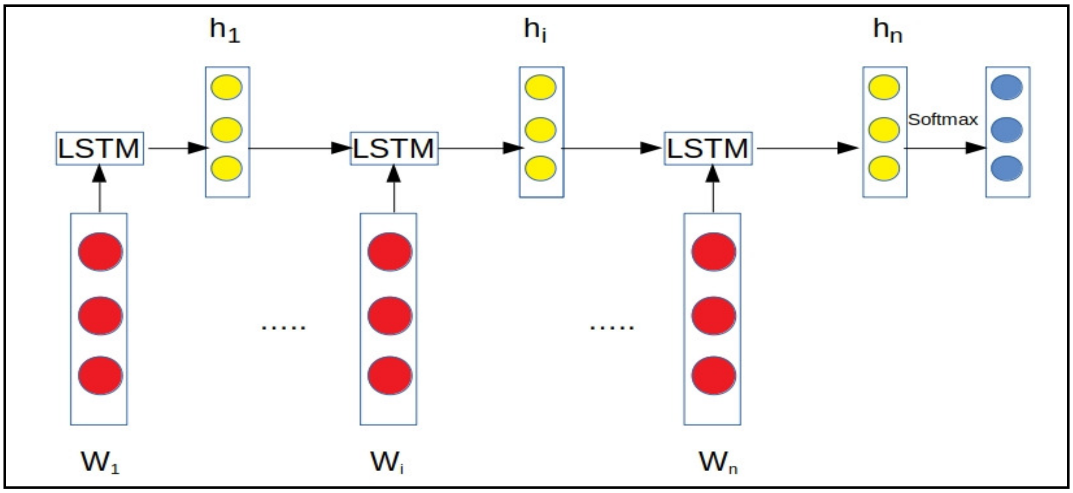

Fig. 1. LSTM recurrent network architecture.

As far as state-of-the-art applications of LSTM are considered, many of the industry giants are using it. Products of Apple (Siri, Quick type), Google (Google Translate, Allo), and Amazon (Alexa) have the usage of LSTM for performing intelligent tasks and learning.

CNN and RNN both can be used for te task with a variety of models. CNNs in general can easily model the local as well as spatio-temporal features. Therefore, they are good for task like computer vision, text processing and speech recognition. Another variant of CNN, called RNN, can easily capture and represent the long term dependencies and is recursive in nature.

The recursive loop make RNNs a better choice in some pattern recognition tasks such as text processing and video processing. We apply the proposed approach on a variety of text classification tasks such as sentiment analysis, customer feedback classification and user rating classification. Experimental results demonstrate the viability of the method in solving these classification problems.

V. PROPOSED METHODOLOGY

The proposed methodology is based on training a Long Short Temporal Memory (LSTM) neural network architecture for text classification. The benefit of using LSTM architecture is that the system can remember what it learned in previous steps. LSTM is a variant of Recurrent Neural Network (RNN). LSTM is relatively insensitive to the gap lengths. This makes it a better choice as compared to other sequence models such as recurrent neural networks (RNN) and hidden Markov models (HMM). In many problems, the depth of the LSTM network is more important than the number of memory units in it. Every extra hidden layer in LSTM add a level of abstraction to it and may understand the problem in a better way. LSTM networks having multiple layers are also called as stacked LSTM networks, which proved to be a better choice for sequence prediction tasks and speech recognition.

Step 1: Pre-processing of the Data

Certain pre-processing steps that need to be performed are:

i. Convert all text to small case

ii. Remove stop words from text

iii. Remove numeric values from the text that have no significance during classification task

iv. Use regular expressions to replace and remove some inadvertent bad symbols

Step 2: Vectorize the text

Step 3: Find out the unique tokens

Step 4: Convert categorical labels in data to the numeric values. In some cases the class labels may be categorical then those need to be converted to numeric values for convenient processing.

Step 5: Train test split

Step 6: Input to the deep learning architecture

Here, a deep learning architecture LSTM layer and other auxiliary layers is used. The architecture can be tweaked based on the number of classes and nature of the data.

Layer 1: In first layer, for a multi-class unstructured text data, we keep length of input vector in the range of N to 10xN, where N is the number of classification classes.

Layer 2: It is a SpatialDropout1D layer that is used to perform variational dropout in text processing models.

Layer 3: It is a recurrent sub-system of LSTM layer with 10xN number of memory units. If the number of classes (N) is having value 10 then LSTM will have 100 memory units.

Layer 4: It is the output layer which produces N output values, one for each of the classes for text classification. We use softmax activation function for output.

Loss Function:

We use cross entropy function as the loss function.

$$\text{Loss}= -\sum_{d}\sum_{n=1}^{N} P_{d}^{g}\,(s).log\big(P_{n}(d)\big) \qquad (1)$$

Where, d is a text fragment (a sentence) such that d∈D and D is the training data set. N is the number of classes in training set, d is a text fragment (a sentence). Term Pn (d) is the probability of fragment d belonging to class n given by softmax function. Pgd (d) is the probability whether n is the correct classification class or not, has value 0 or 1.

VI. EXPERIMENTAL RESULTS

The proposed method has been tested on three different data sets. These data sets have been chosen to include a variety in the task of text classification in terms of number of classes and problem domain of data sets. One data set represents binary classification and other two are for multi-class classification. The data sets represent the problems of sentiment analysis based on movie reviews, literary data set of novels and bug triaging problem from software industry. This variety at different levels is a sufficient and reliable metric for assessing the viability proposed technique. The system has been implemented in Python using keras deep learning library. For training Amazon EC2 cloud instances have been used. The training was performed on individual data sets for 10 epochs. The following data sets have been used to train and test the system:

- **IMDB Movie Review Data Set of 50K Movies [8][9]:** This contains the 50 thousand movie reviews having only two labels positive or negative for classification. The data set is provided by Stanford University for binary sentiment classification. The data set was described in the work of Andrew *et al.,* [1] and available for download at the given link [2].

- Obfuscated Multi-classification Data Set [10]: The data set is available at Kaggle where objective is to train a system that can suggest if an input text belongs to 12 popular novels having indexing from 0-11.

- DeepTriage Data Set [11]: The data set is available at [4]. The data set is used for training the bug triaging process. The triaging refers to automation of assigning a potential developer who can fix an issue of bug. Data set contains the bug reports of Google Chromium (383,104), Mozilla Core (314,388), Mozilla Firefox (162,307) web browsers.

The training and validation accuracy of the proposed technique have been shown in Fig. 2. The average test validation accuracy of the method is 80.90% on the three data set.

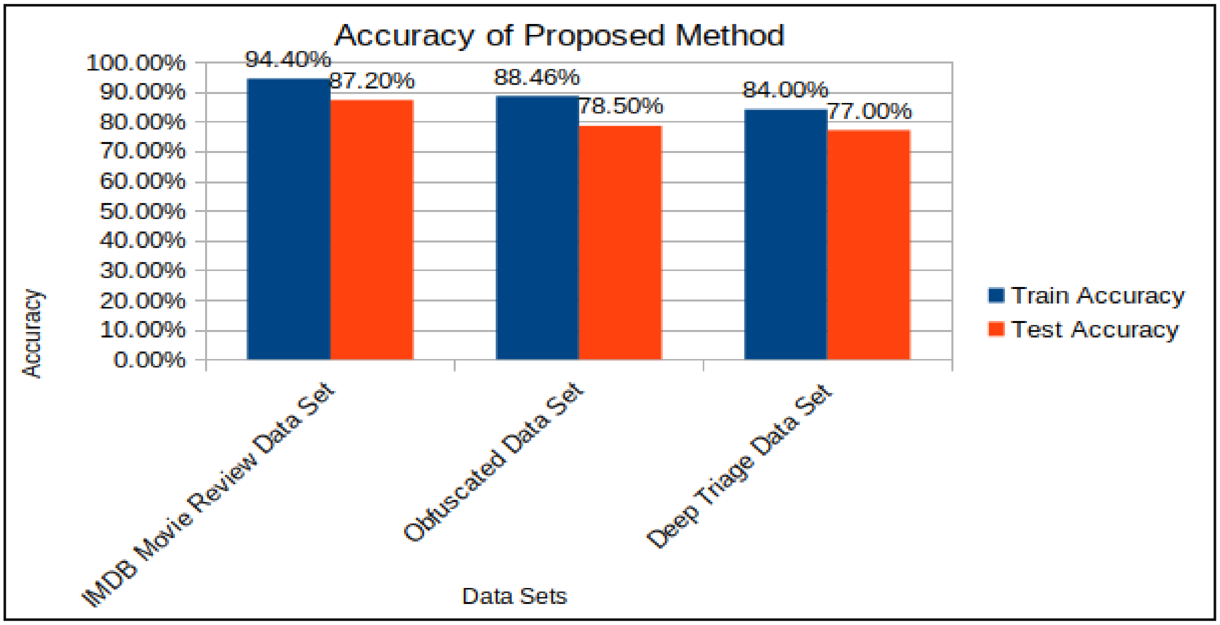

Fig. 2. Accuracy of the proposed method on three different data sets.

Table 1: Performance of Proposed Technique on Three Different Data Sets.

S.No	Data Set	Number of Classes	Sample Size	SVM (NL) Accuracy	CNN Accuracy	Validation Accuracy
1	IMDB Movie Review Data Set [8] [9]	2	50,000	77.40%	81.67%	87.20%
2	Obfuscated Data Set [10]	12	--	66.80%	72.33%	78.50%
3	Deep Triage [11]	—	Google Chromium (383,104), Mozilla Core(314,388), Mozilla Firefox (162,307)	62.00%	73.00%	77.00%

VII. COMPARISON WITH OTHER METHODS

Table 1 shows the comparison of proposed techniques with other two. Three datasets used for comparison are IMDB Movie Review Data Set, Obfuscated Data Set, and Deep Triage Data Set. The proposed method has been compared with other two methods of text classification namely non-linear support vector machine (SVM-NL) and convolutional neural network (CNN). For training of the classifier, Amazon EC2 cloud instances have been used. The training for the proposed method was performed on individual data sets for 10 epochs. The average accuracy of the proposed method is 80.90%.

VIII. CONCLUSION

Classification of voluminous data is a key challenge. Understanding the meaning in data and classifying it in pre-known categories is very useful. It has many potential applications such as sentiment analysis on social media, news categorization, spam filtering, customer feedback & complaint grouping, movie & product reviews, document classification etc. While the traditional approaches fail to scale to large volume and variety of data, deep learning based approaches outperform. Paper has presented a long short term temporal network model for training a deep learning classifier for text classification. Experimental results show that the proposed technique is more accurate than the traditional methods such as Support Vector Machines (SVM). Moreover, it is better than deep learning model of Convolutional Neural Networks (CNN). The average accuracy of the proposed method is 80.90%.

Future research scope in the field belongs to the exploration of newer applications of text classification and improving the classification accuracy further.

REFERENCES

[1]. Adamuthe, A. C., & Jagtap, S. (2019). Comparative Study of Convolutional Neural Network with Word Embedding Technique for Text Classification. *International Journal of Intelligent Systems and Applications*, *11*(8), 56.

[2]. Burkhardt, S. (2018). Online Multi-label Text Classification Using Topic Models (Doctoral dissertation).

[3]. Akhtiamov, O., Fedotov, D., & Minker, W., (2019). A Comparative Study of Classical and Deep Classifiers for Textual Addressee Detection in Human-Human-Machine Conversations. *In International Conference on*

Speech and Computer (pp. 20-30). Springer, Cham.

[4]. Yada, S., Kageura, K., & Paris, C. (2019). Identification of tweets that mention books. International Journal on Digital Libraries, 1-23.

[5]. Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1275-1284). ACM.*

[6]. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research, 2*, 419-444.

[7]. Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[8]. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, & Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

[9]. http://ai.stanford.edu/~amaas/data/sentiment/

[10]. https://www.kaggle.com/alaeddineayadi/obfuscated-multiclassification

[11]. http://bugtriage.mybluemix.net/

How to cite this article: Sharma, S. K. and Sharma, N. K. (2019). Text Classification using LSTM based Deep Neural Network Architecture. *International Journal on Emerging Technologies,* **10**(4): 38–42.