Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 3, July 2021:1665- 1674

Research Article

Rumor Detection Using Various Deep Learning Approaches

Meena Talele
Department of Information Technology
VESIT
Mumbai, India
meena.talele@ves.ac.in

Smita Jangale
Department of Information Technology
VESIT
Mumbai, India
smita.jangale@ves.ac.in

M. Vijayalakshmi
Department of Information Technology
VESIT
Mumbai, India
m.vijayalakshmi@ves.ac.in

Abstract - As the endless development in web 2.0 and ease of access methods, devices upcoming new technologies like Social Media, Mobile, Analytics and Cloud-generates infinite stream of data. The misinformation can spread widely and rapidly in online social network. Due to potential harm this circulate may bring to public, so false rumor detection is demanding and important. Previous studies are mainly based on various machine learning algorithms and deep learning techniques. In this paper, various rumor detection techniques using Deep Learning Models like Long Short Term Memory (LSTM), Convolutional Neural Network (CNN), CNN-LSTM, Bidirectional LSTM (BiLSTM) and CuDNNLSTM(layer with LSTM) on textual data are performed and analysis has been done. These models perform binary classification of tweets into rumors and non-rumors. Comparative Analysis has been done with results on same dataset by existing machine learning algorithms and our deep learning models. Our deep learning models outperforms the baseline machine learning algorithms.

Keywords— Rumor Detection, CNN, LSTM, BiLSTM, CuDNNLSTM, social media, GloveVector.

1. INTRODUCTION

Today, in the world where users on social media are tremendously increasing and conveying their opinions on these platforms, there are chances that many users might be posting rumorous messages over social media platforms for their personal or organizational benefits, which is inappropriate. Thus, to recognize such messages on the social network platforms we need to detect these rumors earlier to avoid its adverse impact on people or organization or any

other aspect for which the rumor was circulated intentionally. A rumor is defined as any piece of information put out in public without sufficient knowledge and/or evidence to support it thus raising a question on its authenticity.
In order to differentiate rumors from true facts people and organizations depends on acuteness and inspecting journalism. There are some rumor debunking websites are available like emergent.info,politfact.com, sina reporting center, but these are based on manual investigating techniques ,these are not exhaustive in tropical coverage and takes long delay for investigation. So there is need to automate this, hence various machine learning and deep learning techniques are used.

The content broadcasted by people on social media can be in the form of text, URL, punctuation, tags and many more to analyze manually. Deep Learning can be used effectively to identify the rumors. This paper involves analyzing results by implementing different deep learning models on the PHEME dataset. These models can be used for recognizing rumors and non-rumors by implementing them on twitter dataset.

In this paper following things are described, section II describes pervious works already performed for rumor detection, and section-III described the models that we have used for achieving the maximum possible accuracy for detecting rumors in minimum possible time. Section-IV performs analysis of results obtained from different models, to determine which model is most suitable for rumor detection.

2. RELATED WORK

Already available rumor detection models prominently applied machine learning techniques along with extracted features from text, user attributes, modes of transmission and time characteristic. Reference [1] used deep bidirectional GRU based neural network for rumor detection.

Reference [13], they trained J48 classifier to carry out characterization of tweets massages which are based upon extracted features. They used Twitter API for extracting tweets, which are of previous 7 days. Reference [12] narrated unsupervised approach for tracking the rumor event and produce real time revisions dynamically based on any extra info obtained.

Arkaita Zubiaga[4] used SVM, Random Forest,Navie Bayes, Maximum Entropy, CRF classifiers on PHEME dataset, which is a twitter dataset which records tweets during five breaking news in social media. Muhammad Zubair Asghar[5] narrates deep learning models BiLSTM with CNN on textual information to classify tweet into rumor and non-rumor(1/0). Reference [6] applied a multitask learning models with a two task varacity+stance and varacity+detection by using branch LSTM and NileTMRG. Reference [8] classified communication among a rumorous social media tweet and a reply tweet as support, deny, query or comment. They used a CNN- neural network using ELMo embeddings of tweet text joined with auxiliary features and gains F1-score of 44.6% and for veracity prediction applied a MLP-Neural network and gains a F1-score of 30.1%. Weiling Chen, Yan Zhang narrated [2] RNN and autoencoders to find rumor anomalies, they proposed some features based on user comments in order to improve rumor detection performance

Reference[9] investigated rumor detection on social media platform Facebook massages- enquiry opinions, they proposed ICDM model which recognizes enquiry messages and apply a rule-based technique which contains regular expression to differentiate the message as enquiry.

Reference [10] narrated different review of already available fake news detection methods, also fake news characterization, feature removal and applying models on some common representative datasets.

3. PROPOSED METHODOLOGY

3.1 Dataset

We used 'PHEME rumor dataset' which was collected by journalist. These rumors are associated with nine different breaking news. It was created for the analysis of social media rumors, and contains Twitter communications which are started by a rumorous messages and the communications include tweets responding to those rumorous tweets.

PHEME dataset consist of several tweets associated with nine different breaking news incidences. Some of them are rumors and some are non-rumors. It consist of total 105392 records (tweets) and each of them are described by five different fields. 'category' specifies whether the tweet is a rumor or non-rumor. 'event' specifies the incidence to which the tweet belongs. 'is_src' specifies whether it is a source tweet or reply tweet. 'text' contains actual textual tweet.

A. Preprocessing

The contents of the 'text' column is noisy so we need to preprocess it. We preprocessed the data by removing all the stop words. Then we have converted all the occurrences of emoji with word describing the meaning of that particular emoji. Later preprocessing also involves removing all the '#s' and urls. Then entire text was converted to lower case then stemming was performed to identify the root words associated with words in tweets. Finally we removed the records with null information.

3.2 One - hot - encoding

The data in the category column has values in the form of rumor/non-rumor. This data is label encoded. This may confuse the model while training. Hence it is one-hot-encoded. In this technique, the category column is divided into 2 different columns for each category. Each column will consist of either 0 or 1 value depending on rumor/non-rumor category.

Category	One-hot encoding
Rumor	[1 0]
Non-rumor	[0 1]

3.3 Word Vectoring

Word embedding is applied on the text column of the tweets. In this process we convert word representations into vectors so that they could be understood by our deep learning models. For word embedding we used GloVe model with 100 dimension word vectors, it is a model for distributed word depiction. The model is an unsupervised learning technique for acquiring vector representations for words. Here words are mapped into relevant space, the distance among words is associated to semantic similarity.

We first transforms all text samples in the dataset into sequences of word indices. A "word index" is an integer ID for the word. Then we prepare an "embedding matrix" which will consist of index i, the embedding vector for the word of index i in our word index. Then embedding matrix is loaded into a Keras Embedding layer, trainable is set to be frozen that means its weights-the embedding vectors, will not be changed during training. Lastly we applied various deep learning models on embedding vectors.

3.4 Padding and Splitting

The length of the tweets is not the same. Therefore, we keep the length of the text sequence equal to 1000 by padding zeros at the beginning (pre-padding). After preprocessing 90 percent data is used for training set and 10 percent is used for validation set.

3.5 LSTM

It is difficult to train the RNN that require long term dependencies to predict the next word in the sentence because of the vanishing gradient problem. To overcome this, we are going to implement a variation of RNN called Long Short Term Memory model (LSTM).

An LSTM layer includes a set of recurrently connected memory blocks. Each block consist of one or more recurrently joined memory cells and three multiplicative units - the input, output and forget gates. These gates supply continuous analogues of write, read and reset operations for the cells. Each gate has valves to control the information which passes through the memory pipeline.

We used LSTM layer of 132 units with dropout value of 0.2 and dense layer of 2 units and sigmoid activation function for binary classification. The model is compiled using adam optimizer having categorical cross-entropy as the value of loss gradient.

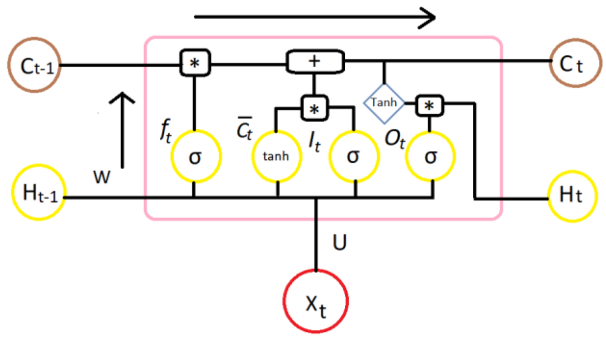

Fig. 1. LSTM

3.6 CNN

CNN is deep, feed-forward artificial neural networks which employ a variation of multilayer perceptrons designed to require small amount of preprocessing.

Text as a sequence is passed as input to a CNN. The embedding matrix is passed to embedding layer. Four different filter sizes are applied to each CNN layer, and MaxPooling1D layers are applied to each layer. All the outputs are then concatenated. Then finally two Dense layers are applied for which first dense layer 'relu' activation function is used whereas for another 'softmax' activation is used. Then the model is compiled using 'rmsprop' optimizer having categorical_cross entropy as the value of loss gradient. model.summary() will print a brief summary of all the layers with their output shapes.

3.7 CNN-LSTM

Text Classification can also be done using Convolution Neural Network (CNN). The advantages of CNN to determine local features and that of LSTM to interpret the sequential information can be combined in the CNN-LSTM model.

The architecture of CNN-LSTM has five Layers. The input layer has a pre-processed input sequence of tweets which is passed on to the embedding layer. The Embedding layer converts the word to its vector representations. The output of this layer is taken as input to the convolution layer. This layer uses 64 filters which performs convolution operation on the embedding matrix and generates feature maps. In convolution operation, element-wise multiplication between the embedding vectors and filter values is performed which is then summed up to obtain one number. This results in one-dimensional feature maps. This is done with the help of the Rectified Linear Unit (ReLU) activation function. These maps are of variable sizes on which max-pooling operation is performed in the pooling layer. In max-pooling, we extract the largest number of values from each feature map. This helps in reducing the overfitting. The pooling layer is followed by the LSTM layer with 32 units. Finally there is a dense layer with sigmoid activation function. Then the model is then compiled using 'adam' optimizer having categorical_cross entropy as the value of loss gradient

3.8 BiLSTM

Bidirectional LSTM can get improved model outcome on sequence classification issue. Bidirectionl lstm is used where full timestamp of input sequence is accessible. With BiLSTM we feed the learning algorithm with the original data once from beginning to the end and once from end to beginning. This network can results into improved and faster results, apply fuller learning in network.

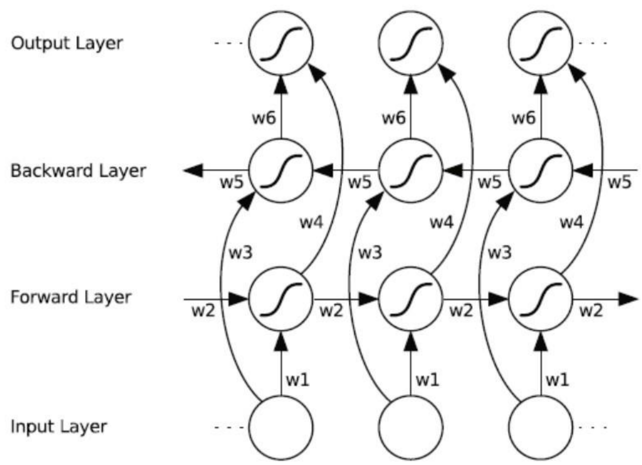

Fig. 2. BiLSTM

Our model involves first starting with embedding layer which takes preprocessed tweets, then followed by Dropout layer of value 0.2. Further we apply two Bidirectional LSTM layers followed by dropout layer of value 0.25. Finally there is a dense layer with sigmoid activation function. Then the model is then compiled using 'adam' optimizer having categorical_cross entropy as the value of loss gradient

3.9 CuDNNLSTM

We used the CuDNNLSTM layer in BiLSTM model which uses a Deep Neural network using CUDA (Compute Unified Device Architecture). CuDNN library aids in fast processing with the help of parallel processing GPU. The network has four layers, Embedding layer which helps in creating the 100-dimensional embedding vector, Dropout layer of value 0.2, CuDNNLSTM layer of 64 units, CuDNNLSTM layer of 32 units Dropout layer of value 0.25, and Dense layer of 2 units and Sigmoid activation function for binary class output. The model is then compiled using Adam optimizer having categorical cross-entropy as the value of the loss gradient.

3.10Experimental Matrices

To evaluate the performance of the models on PHEME dataset we used the evaluation indicators such as precision, recall, F1-score and accuracy using following formulas:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + True\ Negetive} \quad (2)$$

$$F_1 score = 2 . \frac{Precission\ .Recall}{Precission + Recall} \quad (3)$$

$$Accuracy = TP+TN/TP+FP+FN+TN \quad (4)$$

4. EXPRIMENTAL RESULTS and ANALYSIS

We used paperspace Gradient which is cloud Based GPU service on Keras environment with Jupyter notebook for training, development and deploying deep learning model. We applied deep learning models on complete 9 events of PHEME dataset.

Previous studies [4] used machine learning based approach which extracts social, content and social+content feature extraction on dataset of 5 events of PHEME ,their results are

Features	Content			Social			Content+social		
Classifier	P	R	F1	P	R	F1	P	R	F1
SVM	35.5	44.5	39.5	33.7	52.4	41	33.7	48.3	39.7
Random Forest	27.1	8.7	13.1	34.3	43.3	38.2	27.5	9.9	14.5
Naïve Bayes	30.9	72.3	43.3	29.4	1	2	31	72.3	43.4
Maximum Entropy	32.9	42.5	37.1	33.6	47.6	39.4	33.8	44.2	38.3
CRF	68.3	54.5	60.6	46.2	26.8	33.9	66.7	55.6	60.7

Table 1: State of the art Baseline Result with ML

We applied LSTM, CNN, CNN+LSTM and BiLSTM models
With word embedding and without word embedding

Table consist of training and validation performance results obtained after 10 epoch without word embedding.

	Models	LSTM	CNN	CNN-LSTM	BiLSTM	CuDNNLSTM
Training	Loss	43.36	24.34	29.72	36.06	41.33
	Accuracy	78.83	88.63	86.11	82.97	80.24
	F1	77.6	88.62	69.11	82.02	79.88
	Precision	77.5	88.62	88.55	84.35	80.74
	Recall	77.75	88.62	56.89	79.83	79.05
Validation	Loss	58.93	97.8	80.11	72.35	62.23
	Accuracy	73.31	69.62	69.41	72.38	72.89
	F1	71.95	69.6	58.04	71.58	72.63
	Precision	72.46	69.6	73.04	73.3	73.51
	Recall	71.49	69.6	48.24	69.94	71.77

Table 2: Experiment Results without word embedding with DL

	Models	LSTM	CNN	CNN-LSTM	BiLSTM	CuDNNLSTM
Training	Loss	53.86	31.12	49.15	53.06	51.77
	Accuracy	73.46	85.6	76.06	74.12	74.72
	F1	72.27	85.6	73.09	75.01	74.78
	Precision	72.23	85.6	75.61	70.82	74.41
	Recall	72.35	85.6	70.79	79.79	75.17
Validation	Loss	53.68	85.76	54.86	53.81	53.25
	Accuracy	73.66	70.08	73.83	73.66	74.03
	F1	72.89	70.06	71.98	74.68	74.18
	Precision	72.85	70.06	73.82	71.42	74.05
	Recall	72.96	70.06	70.26	78.26	74.3

Table 3: Experiment Results with word embedding with DL.

Following figure is a chart which depicts the validation accuracy of different models from 1 to 10 epoch with word embedding.

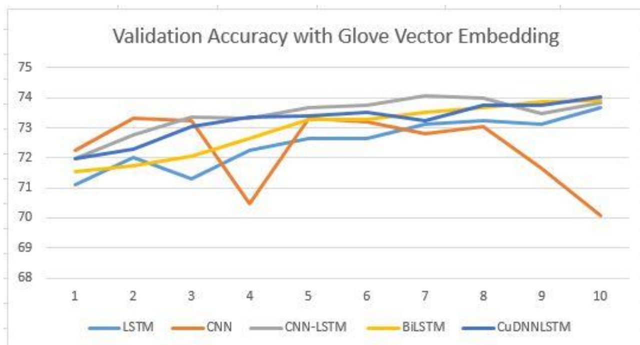

Fig. 3. Validation accuracy chart for 10 epoch with Glove Vector Embedding

Following figure is a chart which depicts the validation accuracy of different models from 1 to 10 epoch without glove vector embedding.

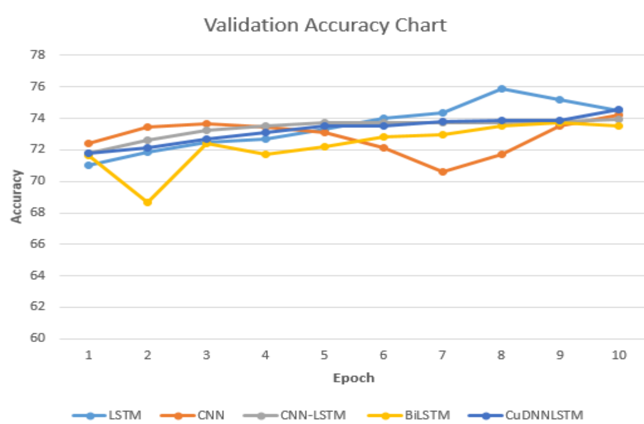

Fig. 4. Validation Accuracy chart for 10 epoch

5. CONCLUSION AND FUTURE SCOPE

Thus, we have analyzed the efficiency and accuracy of different deep learning models for rumor detection using PHEME dataset. Here we used complete 9 events of PHEME dataset for rumor detection. Result obtained from

BiLSTM and CuDNNLSTM are almost similar, but on CuDNNLSTM model training is very much faster and less time for training as compare to BiLSTM model.

Ml algorithms linear regression, logistic regression, random forest, decision tree etc. can learns using data and improve on their own to get results and are linear in operation.

DL algorithms has large computation power, provide automatic self-improvement and automatic feature extraction. DL can process large amount of data with high accuracy. They takes longer time for training, also don’t required human involvement.

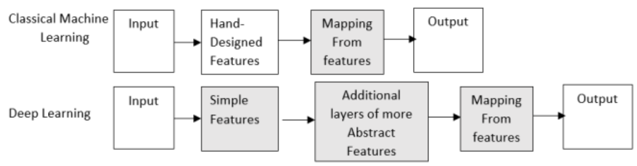

Fig. 5. Machine Learning Vs Deep Learning

Our proposed DL based models sufficiently outperforms as compare to baseline ML classifiers.

This rumor detection using deep learning can further also be helpful for journalist who spend a lot of hours in physically determining whether the message is rumor or a true fact.

In this research, only text-based features are used for rumor/non-rumor binary classification, whereas adding advance types of the features which may results more efficient, sound results. This research concentrated only on the English text representation.

Along with text-based features, different types of features, such as social context and images, propagation based features can be investigated for obtaining more efficient results. More experiments can be conducted on the textual data.

In future, we can explore other deep learning techniques such as developing hybrid models for rumor-classification, veracity prediction and detection.

6. REFERENCES

[1] Lizhao Li, Guoyong Cai, Nannan Chen, “A Rumor Events Detection Method Based on Deep Bidirectional GRU Neural Network,” in 3rd IEEE International Conference on Image, Vision and Computing, 2018.

[2] Weiling Chen , Yan Zhang ,Chai Kiat Yeo , Chiew Tong Lau ,BU Sung Lee “Unsupervised rumor detection based on users’ behaviors using neural networks “ ,Science Direct , pages 226-223,1st April 2018 .

[3] Mohammed Al-Sarem ; Wadii Boulila ; Muna Al-Harby ,Junaid Qadir,Abdullah Alsaeedi,“Deep Learning-Based Rumor Detection on Microblogging Platforms: A Systematic Review”,vol 7, IEEE Journal 16 Oct 2019.

[4] Arkaitz Zubiaga, Maria Liakata, Rob Procter University of Warwick, Coventry, UK, “,Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media”, arXiv:1610.07363v1 [cs.CL] 24 Oct 2016.

[5] Muhammad Zubair Asghar, Ammara Habib, Anam Habib, Adil Khan, Rehman Ali & Asad Khattak ," Exploring deep neural networks for rumor detection Journal of Ambient Intelligence and Humanized Computing (2019) https://link.springer.com/article/10.1007%2Fs12652-019-01527-4

[6] ElenaKochkina1, Maria Liakata1, Arkaitz Zubiaga,"All-in-one:Multi-task Learning for Rumour Verification", arXiv:1806.03713v1 [cs.CL] 10 Jun 2018.

[7] Changhe Song, Cunchao Tu, Cheng Yang, Zhiyuan Liu, Maosong Sun, "CED: credible early detection of social media rumors.", Journal of latex class files, vol. 14, no. 8, august 2015

[8] Ipek Baris,and Lukas Schme lzeisen,and Steffen Staab, "CLEARumoratSemEval-2019 Task7: ConvoLving ELMo Against Rumors" arXiv:1904.03084v1 [cs.CL] 5 Apr 2019

[9] Rehana Moin, Zahoor-ur-Rehman, Khalid Mahmood, Mohammad Eid, Alzahrani, Muhammad Qaiser Saleem. "Framework for Rumors Detection in Social Media". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5, 2018 ,Page 439 www.ijacsa.thesai.org ,

[10] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang , and Huan Liu ,"Fake News Detection on Social Media: A Data Mining Perspective", ACM SIGKDD Explorations Newsletter, 2017

[11] [Feixiang Wang,ManLan,Yuanbin Wu ,"ECNU at SemEval-2017 Task 8: Rumour Evaluation Using Effective Features and Supervised Ensemble Models", 51151201049@stu.ecnu.edu.cn, {mlan, ybwu}@cs.ecnu.edu.cn

[12] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, Sameena Shah , "Real-time Rumor Debunking on Twitter" Conference Paper · October 2015 DOI: 10.1145/2806416.2806651