DETECTION OF CREDIBILITY OF ONLINE MEDIA POSTS USING DEEP NEURAL NETWORK

Ravi Ranjan Mallik

Abstract

Nowadays, the Internet and social media have become one of the primary medium to share the information. Every day, we come across hundreds of online media posts and blogs. This paper shows an approach to detect the credibility of the posts and the sources of the post over online media. In this letter, we have performed posts detection method based on one of the artificial intelligence algorithms – Deep Neural Network. While there exist different other mechanisms for the similar work, we have obtained that the proposed method, for this given large labeled news dataset, works better.

Introduction

The News media is one of the most important forms to share news to the audiences and has widened its reach through different mediums: either printed, broadcast or through the Internet. Due to the vast influence of the Internet and ease of access of the reader, the online media posts and blogs have become one of the main sources for news broadcasting. The news through these posts and blogs spread real fast across the web world. Since most of the blogs and posts are produced by social or self-journalist, most of the time, they are biased. These mass-media play a vital role in influencing society. As the news goes viral, it's always challenging to determine whether they or their sources are genuine or fake. Most of the time, there's someone who can take advantage of these fake news and manipulates the public opinion over a certain matter. These posts appear most frequent before or during the election year.

There are several articles proposed describing the detection of such posts. In [9] the authors have described their method for fake news detection using Naive Bayes Classifier. In [6] the author have developed a method using Random Forest algorithm and classifying on the basis of different features. In [5] the authors provide a novel approach to define available techniques for the matter.

In this paper, we have performed the classification of those posts on the basis of their credibility is. With the amount of data increasing every day we have proposed, an artificial intelligence algorithm - Deep Learning approach for the training the model for the detection of credibility of the post. The aim of the study is to examine how this particular method works for this particular problem given a labelled news dataset. The difference between these articles and articles on the similar topics is that in this paper deep neural network along with the concept of word embedding and LSTM was to obtain the credibility of those posts. It was later, tested on a new data set, which gave an opportunity to evaluate its performance on a recent data.

We have used the concept of word embedding and LSTM to the deep neural network model to enhance the proposed model performance. Rest of the article is organized as follows:

Section II will present an overview of the domain fields and its features that are employed in the proposed work. Section III describes the detailed methodology followed for the proposed work using the techniques those are mentioned in section II. Section IV will visualize the result obtained on the selected dataset using proposed work and in Section V will discuss what modification could be performed in near future.

Deep Neural Network

This As defined in [7], “Deep learning, as a sub-class of machine learning, attempts to learn in multiple levels, corresponding to different levels of abstraction of information using artificial neural networks.” The model trains itself by abstracting the relevant feature from the input dataset by the help of the parameter that we provide during the compiling and training of the model. Backtracking the past, we found that it has made a vast impact in the area of data processing, whether it's image, text or audio dataset.

In the domain of Natural Language Processing, - a subclass of text processing, Deep learning methods is being deployed to perform tasks such as language modelling, part-of-speech tagging, machine translation, named entity recognition, sentiment analysis, and paraphrase detection without external hand-designed resources or time-intensive feature engineering. It develops and uses “embedding,” which refers to the representation of symbolic information in natural language text at word-level, phrase-level, and even sentence-level in terms of continuous-valued vectors

LSTM

First introduced in 1999 [3], this feature of RNN bypasses the vanishing gradient problem during training by isolating the memory unit and the output unit where the current memory unit linearly depends upon the previous memory unit in a chained process. This concept, Learning-to-forget, learns from the past behaviour of the input set, keeping the track of previous input which was the main drawback of traditional RNN. It's architecture composed of three gates: input gate, providing set of input to neuron; the forget gate to drop the unwanted information from input set based on previous input and the output gate to represent the expected output, it enhances the performance of the model. As proposed in [10], we found that word embedding suited LSTM than one-hot encoding as the text index in word embedding a lot more than it's alternatives which help in training the model with an ease.

Word Embedding

As in paper[13], the Deep Neural network Embedding trains the network from high dimensional data space to low dimensional data space of embedding by minimizing both, the objective function of dimensionality reduction algorithm and the reconstruction error of the input. It bypasses the “out-of-sample” problem which occurs in one hot encoding for larger datasets.

Methodology

In this paper, the data is being processed with the deep neural network models with Embedding and LSTM, where Word Embedding overcomes the out-of-sample problem and LSTM controls the loss of relevant information while training.

The proposed method is achieved in five steps as shown in Fig.[1] which has been followed in book [4]:

Data Preprocessing

The major challenge is to find out dataset of the posts or blogs on which detail analysis is to be performed and here, we have taken the labelled dataset from the Kaggle[8], collected from different stream pages. From the available 20 fields, we have selected some relevant fields to retrieve news articles as our input dataset. With over 14000 records to work on, we have split the data into train and test in the ratio of 0.2 (train data: 80%, test data: 20%). Further, we have performed the steps of data preprocessing techniques over both, training dataset and test dataset. Since the input to our model is only numeric tensor values, the data preprocessing on these dataset are carried on these set of steps:

Data Cleaning

In this step, we have parsed the data collected from the different stream and from the obtained CSV file, we have abstracted the relevant fields to numpy array. for our model, we have selected the post's text and its label to differentiate them in according. Further, we have ignored the data having null values to reduce training accuracy. Later, we have used the parsed dataset with no missing values for the vectorization step.

Vectorization

In the step, we have converted the parsed input text data from the previous step into numeric tensors which will behave as inputs to the Deep Learning Model. After we vectorize the input text, we have further, passed those vectors to the model for training purpose which achieved in two steps; First, we performed Tokenization, where we split the test string are into n-grams chunks and, then we create their associated numeric vectors using the word embedding concept, where we have embedded a predefined word embedding Global Vectors for Word Representation (GloVe)[4].

Embedding Preprocessing

Deep neural network model requires a large dataset to learn powerful features, in failing in feeding that, we have here used a predefined word embedding stacked to the model through Embedding Layer. As the predefined embedding is an archive, to add it to model, we have followed two steps[1]:

Parse the file into map index; Here, we have created a key-value pair for each of the indices present in the word embedding to create the mapping sample for our text input tensors which further behaves as an associate to the numeric tensors while training

Construct an embedding matrix to pass it to the layer; After we created a mapping file for input data, now we have constructed a matrix with the same dimensions as of our input dataset. ie.

$$embedding\_matrix=[max,dim] \qquad (1)$$

This matrix has a shape of max_words (m) from the input dataset that we are using and embedding_dimension (d), that the key-value pair can acquire. The size of the matrix is as same as the size of our dataset, thus there won't be an input-output mismatch

Building the Network Model

In this section, we have described our proposed network model for classification.

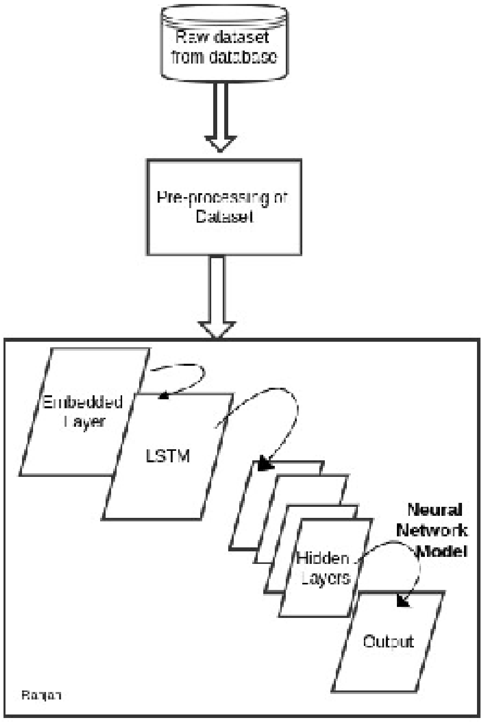

Figure 1. Project Work flow Model

The network architecture

It is the core building block of the network model. And our proposed model follows the Sequential Network Model Architecture, where each node in a layer is connected to each node of the adjacent layer.

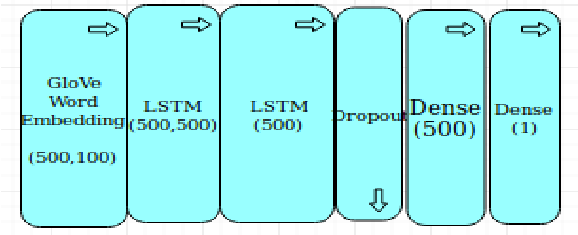

Figure 2. Proposed Network Layered Model

As in fig[2], our model is composed of four different layers; Embedding Layer, LSTM Layer Dense, Layers and Dropout Layer.

The activation function

The activation function, also known as tensor operations, defines the behaviour of output from the node for the given input or set of inputs. When activated, it helps the neural network model to reduce the learning transformation of tensors by deciding which information from the input is relevant by transforming them non linearly.

$$output = activation(dot(input, kernel) + bias) \qquad (2)$$

where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer.

With different types of activation function available, our network model has used :

ReLU

ReLU or Rectified Linear Unit function, defined as:

$$F(x) = x^+ = max(0,x) \qquad (3)$$

where x is the input, is a non-linear function that back propagates the errors. Depending on the input behaviour, relevant or irrelevant, it activates the neurons making the network parse and easy to compute.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \qquad (4)$$

Sigmoid function

This function is preferred when the model is expected to generate binary output or in the range[0,1].

Mathematically,

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}. \qquad (5)$$

where x is a large set of output from the previous layer. We have used the Sigmoid function in our output layer of the network for the binary classification of the posts.

Compiling the Network Model

In this section, the compilation of the proposed network model is performed based on two primary factors:

Loss function

This function is employed to minimizes the loss in quality of the selected set of parameters based on their induced score with ground truth labels, and for our dataset, we have chosen binary cross-entropy to obtain the probabilistic output.

Binary cross-entropy is considered to be an ideal loss function while using the Sigmoid function in the output layer and expected output is either in the range of 0 and 1.

Optimizer

The optimizer reduces the different cost function that determines the update of the network based on the loss function. For the proposed model, we have chosen RMSprop algorithm for deep learning model.

RMSprop has proven its effectiveness as a modification of Adagrad algorithm by discarding the extreme past history. The exponentially decaying average and the learning rate of 0.001 helps to optimize the model. It minimizes the loss during training due to loss function and mini-batch samples.

Classification of posts

For the better distribution, we made random shuffle to the dataset, and after that, we divided them into two subsets: training dataset and test dataset and further the training dataset is partitioned into training and validation dataset. The neural network will then use the training dataset for training classifier model whereas it will use validation dataset for tuning some global parameters of the classifier. The model then uses the test dataset to estimate how well the model performs on new data.

At this final module, we are going to feed the test dataset to the trained model to analyze the prediction accuracy of the model. Based on obtained accuracy, we are going to evaluate different classification parameters and if possible, even enhancing the accuracy.

Results and Discussion

We randomly shuffled our dataset and spitted it in ratio of 0.2 as our training and text dataset and tested on our model. In the experiment we found that, model have achieved an accuracy of 95.4%.

The figure [3] show that how the training and test data have performed over the epochs count of 6.

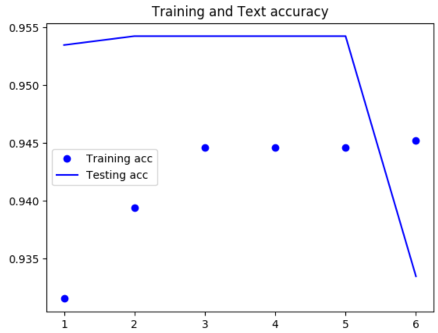

Figure 3. Training and Testing Accuracy

Acknowledgments

The authors are thankful to IJATER Journal for the support to develop this document.

References

[1] Jason Brownlee (Feb 2018) How to Use Word Embedding Layers for Deep Learning with Keras, Available at: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/ (Accessed: 18th November 2018).

[2] J. Hill, M. Jeffrey Pennington, Richard Socher, Christopher D. Manning (2014) *GloVe: Global Vectors for Word Representation*, Available at: *https://nlp.stanford.edu/projects/glove/* (Accessed: Nov 2018).

[3] Felix A. Gers, Jiirgen Schmidhuber,Fred Cummins (1999) 'Learning t o Forget: Continual Prediction with LSTM', *Artificial Neural Networks, 7 10 September 1999, Conference Publication No. 470 IEE 1999*, (), pp. 850-855.

[4] FRANÇOIS CHOLLET (2018) *Deep Learning with Python*, ISBN 9781617294433 edn., 20 Baldwin Road PO Box 761 Shelter Island, NY 11964: Manning Publications Co..

[5] G. Shanmugasundaram (2017) 'Investigation on Social Media Spam Detection', *International*

Conference on Innovations in information Embedded and Communication Systems (ICIIECS), 17(978-1-5090-3294-5), pp. .

[6] Julien Fontanarava, Gabriella Pasi,Marco Viviani (2017) 'Feature Analysis for Fake Review Detection through Supervised Classification', *International Conference on Data Science and Advanced Analytics*, 17(978-1-5090-5004-8), pp. 658-666.

[7] Li Deng and Dong Yu (2014) *Deep Learning: Methods and Applications*, Volume 7 Issues 3-4, ISSN: 1932-8346 edn., Tokyo: Foundations and Trends in Signal Processing.

[8] Megan Risdal (2016) *Getting Real about Fake News*, Available at: *https://www.kaggle.com/mrisdal/fake-news/home* (Accessed: Nov 2018).

[9] Mykhailo Granik, Volodymyr Mesyura (2017) 'Fake News Detection Using Naive Bayes Classifier', *First Ukraine Conference on Electrical and Computer Engineering*, 17(978-1-5090-3006-4), pp. 900-903.

[10] Suman Ravuri,Andreas Stolcke (2015) 'A COMPARATIVE STUDY OF NEURAL NETWORK MODELS FOR LEXICAL INTENT CLASSIFICATION', *ASRU*, 15(978-1-4799-7291-3), pp. 368-374.

[11] Sydney A. Barnard, Soon M. Chung, Vincent A. Schmidt (2017) 'Content-based Clustering and Visualization of Social Media Text Messages', *International Conference on Data and Software Engineering (ICoDSE)*, 17(978-1-5386-1449-5), pp. .

[12] Vivek Yadav (Jun 23, 2017) *Deep learning setup for Ubuntu 16.04: Tensorflow 1.2, keras, opencv3, python3, cuda8 and cudnn5.1*, Available at: *https://medium.com/@vivek.yadav/deep-learning-setup-for-ubuntu-16-04-tensorflow-1-2-keras-opencv3-python3-cuda8-and-cudnn5-1-324438dd46f0* (Accessed: 18th November 2018).

[13] Yan Huang, Wei Wang, Liang Wang, Tieniu Tan (2014) 'A General Nonlinear Embedding Framework Based on Deep Neural Network', *22nd International Conference on Pattern Recognition*, 14(1051-4651), pp. 732-737.

Biographies

RAVI RANJAN MALLIK received the B.E. degree in Information and Technology Engineering from the Rajiv Gandhi Technical University, Bhopal, MP, in 2014, and is currently pursuing the M.Tech. degree in Computer Science and Technology Engineering from the Andhra University, Visakhapatnam, AP. His interest and research areas include text mining, artificial intelligence and deep learning. He may be reached at
raviranjanmallikbe@gmail.com