Indian Institute of Information Technology Vadodara
Government Engineering College, Sector-28, Gandhinagar, Gujarat - 382028

Design Project Report

on

FAKE NEWS DETECTOR

Submitted by

Vanshika Panwar (201851140)
Vikas Saini (201851143)
Shashank Yadav (201851119)
Mansi Gupta (201852019)

under the supervision of

Dr. Bhupendra Kumar

Abstract

An intense inspection the present tweets demonstrates that false news spreads frequently through human than a genuine news does Lie gets traveled around us quicker, and more extensively than reality in all spheres of information, and the effects were more dangerous and horrifying. There are several kinds of tweets like issues on a government, trending topics around the world, mental abuse, urban legends, occasions in calamities. What's more notifying is that it's not just bots that is outpouring the majority of the misrepresentations, studies claimed. It was some specific individuals performing a large share of this crime. Normally general users, as well, they explained. In this case, verified users and those with numerous fans were not more often the center in spreading misinformation of the corrupted posts. Fake news on social media which got viral like a rocket in no time can cause much havoc to our society human and country

I. Introduction

Fake news and hoaxes have been there since before the advent of the Internet. The widely accepted definition of Internet fake news is: "fictitious articles deliberately fabricated to deceive readers". Social media and news outlets publish fake news to increase readership or as part of psychological warfare. The purpose of the work is to come up with a solution that can be utilized by users to detect and filter out sites containing false and misleading information.

We are working on a project on Fake News Detection in machine learning. Right now, we are living in a world of misinformation and fake news. Goal of our project is to detect fake news. In this we will be using basic python programming and basic machine learning.

II. Literature Survey

To facilitate research in fake news detection on social media, in this survey we will review two aspects of the fake news detection problem: characterization and detection. We will first describe the background of the fake news detection problem using theories and properties from psychology and social studies; then we present the detection approaches. Our major contributions of this survey are summarized as follows:

We discuss the narrow and broad definitions of fake news that cover most existing definitions in the literature and further present the unique characteristics of fake news on social media and its implications compared with the traditional media.

We give an overview of existing fake news detection methods with a principled way to group representative methods into different categories.

We discuss several open issues and provide future directions of fake news detection in social media.

III. THE PRESENT INVESTIGATION

In this article, we present an overview of fake news detection and discuss promising research directions. The key motivations of this survey are summarized as follows:

• Fake news on social media has been occurring for several years; however, there is no agreed upon definition of the term “fake news”. To better guide the future directions of fake news detection research, appropriate clarifications are necessary.

• Social media has proved to be a powerful source for fake news dissemination. There are some emerging patterns that can be utilized for fake news detection in social media. A review on existing fake news detection methods under various social media scenarios can provide a basic understanding on the state-of-the-art fake news detection methods.

• Fake news detection on social media is still in the early age of development, and there are still many challenging issues that need further investigations. It is necessary to discuss potential research directions that can improve fake news detection and mitigation capabilities.

We know that there are two classes in a news, whether it’s real or fake one. From the idea of psychological research on false news, we find out the word lengths in a tweet statement can be a great feature as unauthenticated news content a lot of title, words and fictional statements so we added a new feature word length which is actually the count of words in a tweeted statement.

A) Preparing the data set

- Imported our dataset using pandas.
- Perform data cleaning using nltk library by removing stop words.
- Data visualization using libraries like matplotlib, wordCloud etc.

B*)* Tokenization and Padding

- *Tokenize our training data*

This is straightforward; we are using the TensorFlow (Keras) Tokenizer class to automate the tokenization of our training data. First we create the Tokenizer object, providing the maximum number of words to keep in our vocabulary after tokenization.

- *Get our training data word index*

A by-product of the tokenization process is the creation of a word index, which maps words in our vocabulary to their numeric representation, a mapping which will be essential for encoding our sequences.

- *Encode training data sentences into sequences*

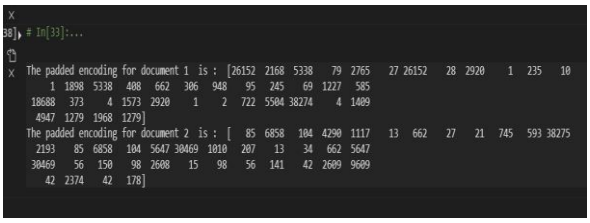
Now that we have tokenized our data and have a word to numeric representation mapping of our vocabulary, let's use it to encode our sequences. Here, we are converting our text sentences from something like "My name is Matthew," to something like "6 8 2 19," where each of those numbers match up in the index to the corresponding words. Since neural networks work by performing computation on numbers, passing in a bunch of words won't work.

- *Get max training sequence length*

We needed to have a maximum sequence length for padding our encoded sentences. We could set this limit ourselves, but in our case we will simply find the longest encoded sequence and use that as our maximum sequence length.

- *Pad the training sequences*

As mentioned above, we need our encoded sequences to be of the same length. We just found out the length of the longest sequence, and will use that to pad all other sequences with extra '0's at the end ('post') and will also truncate any sequences longer than maximum length from the end ('post') as well. Here we use the TensorFlow (Keras) pad sequences module to accomplish this.



C) **Algorithm used**

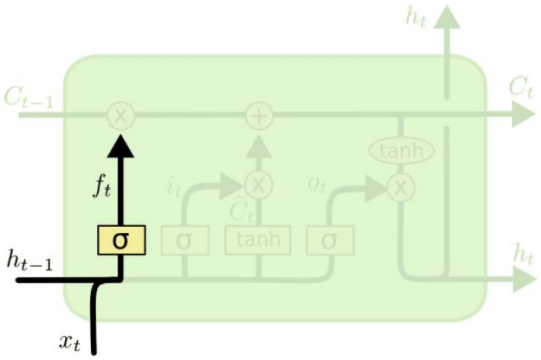
- We have used bidirectional Long Short Term Memory (LSTM) to build our model. There are many advantages of using LSTM over Recurrence Neural Network (RNN).
- RNN are not capable of handling long-term dependencies, while on the other hand LSTMs are explicitly designed to avoid the long-term dependency problem.
- The repeating module in a standard RNN contains a single layer. LSTMs also have this chain like

structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, which interact in a very special way.

- The key to LSTMs is the cell state. It runs straight down the entire chain, with only some minor linear interactions.
- As the cell state goes on its journey, information get's added or removed to the cell state via gates. They contains sigmoid activations. The network can learn which data is not important therefore can be forgotten or which data is important to keep.
- We have three different gates that regulate information flow in an LSTM cell. A forget gate, input gate, and output gate.

a) Forget gate

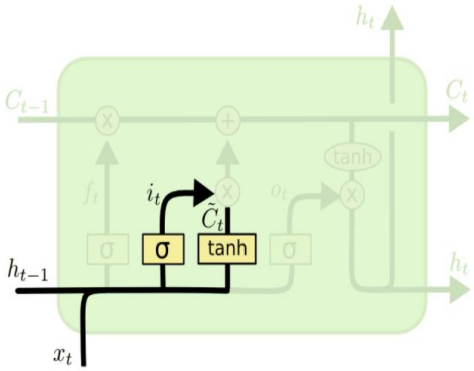
This gate decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function. Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

b) Input gate

To update the cell state, we have the input gate. We pass the previous hidden state and current input into a sigmoid function. It decides which values will be updated by transforming the values to be between 0 and 1..Then the hidden state and current input is passed into the tanh function to squish values between -1 and 1 to help regulate the network. Then it is multiplied by the tanh output with the sigmoid output. The sigmoid output will decide which information is important to keep from the tanh output.

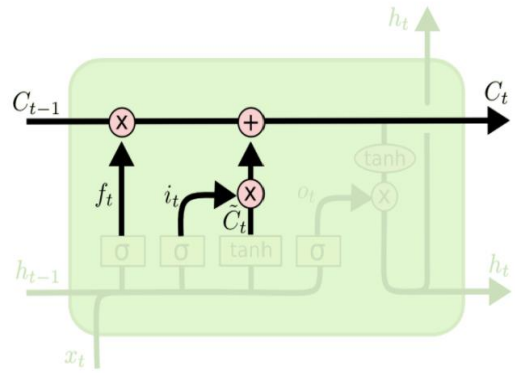


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

c) *Cell state*

First, the cell state gets pointwise multiplied by the forget vector. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then we take the output from the input gate and do a pointwise addition which updates the cell state to new values that the neural network finds relevant. That gives us our new cell state.



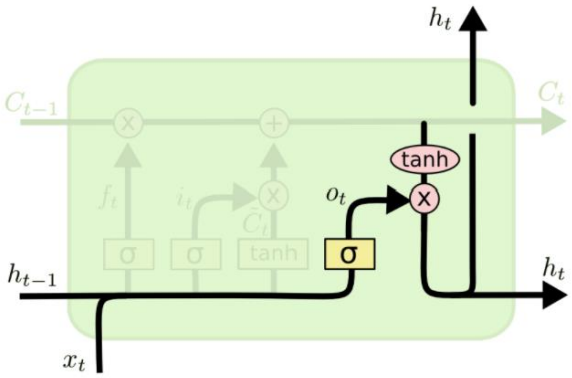
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

d) Output gate

The output gate decides what the next hidden state should be. First, we pass the previous hidden state and the current input into a sigmoid function. Then we pass the newly

modified cell state to the tanh() function. We multiply the tanh() output with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state.

The new cell state and the new hidden is then carried over to the next time step.

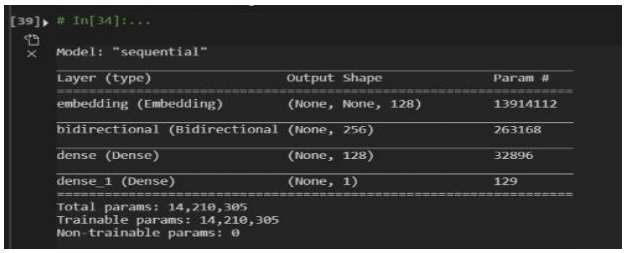


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

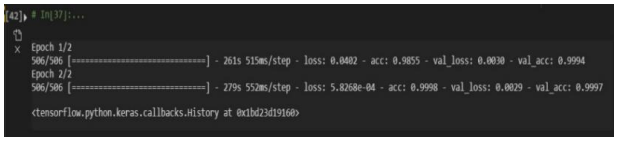
D) Model

- We have used keras API to build our sequential model. Sequential model is a simple stack of layers. First, we have instantiated our Sequential model object and then, we have added layers to it one by one using the add method.
- Our first layer is embedding layer, then a Bidirectional LSTM layer, then a dense layer in which we have used Rectified Linear Unit (ReLU) activation function and finally the output layer which is also a dense layer in which sigmoid activation function is used.



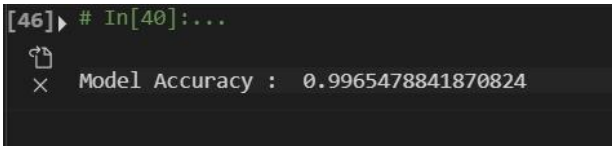
```
[39]▸ # In[34]:...
      Model: "sequential"
      _________________________________________________________________
      Layer (type)                 Output Shape              Param #
      =================================================================
      embedding (Embedding)        (None, None, 128)         13914112
      _________________________________________________________________
      bidirectional (Bidirectional (None, 256)               263168
      _________________________________________________________________
      dense (Dense)                (None, 128)               32896
      _________________________________________________________________
      dense_1 (Dense)              (None, 1)                 129
      =================================================================
      Total params: 14,210,305
      Trainable params: 14,210,305
      Non-trainable params: 0
```

- Once the model is created, we have configured our model with losses and metrics with model.compile() and trained the model with model.fit().



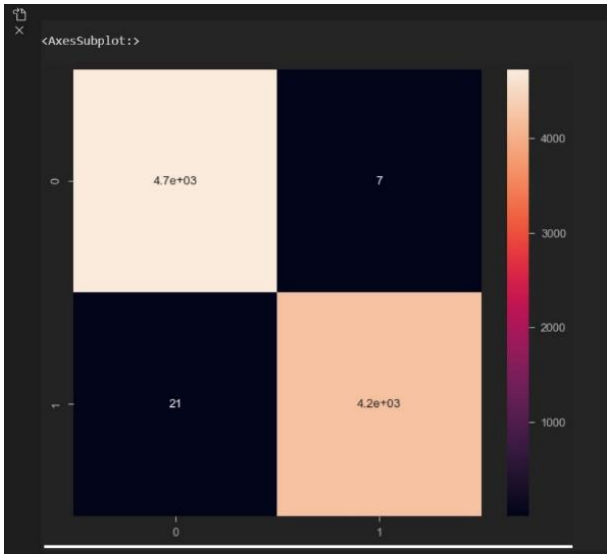
```
[42]▸ # In[37]:...
      Epoch 1/2
      506/506 [==============================] - 261s 515ms/step - loss: 0.0402 - acc: 0.9855 - val_loss: 0.0030 - val_acc: 0.9994
      Epoch 2/2
      506/506 [==============================] - 279s 552ms/step - loss: 5.8268e-04 - acc: 0.9998 - val_loss: 0.0029 - val_acc: 0.9997
      <tensorflow.python.keras.callbacks.History at 0x1bd23d1916b>
```

- We have divided data into train and split and will be using 20 percent data for training and the rest 80 percent for testing.
- The accuracy of the model can be predicted using accuracy metrics.
- Now the model can successfully predict fake or real news from our dataset with accuracy of 0.99654788418.



```
[46]▸ # In[40]:...
      Model Accuracy :  0.9965478841870824
```

IV. Results and Discussions



All codes are written in Python 2.7, using TensorFlow r0.12. All experiments have been performed on a Core™ processor Intel® CPU i7-4790U 3.60 GHz with 16 GB RAM. The dataset utilized in this study is obtained from open Machine Learning Repository accessible at Kaggle. Each news article in dataset consists of title, text, subject

and date content. The vocabulary size for both the dataset is approximately 111 MB. To train the models Root Mean Square prop (RMSProp) and Adam adaptive learning rate algorithms are chosen. RMSProp is unpublished adaptive learning optimization algorithm that clips the gradients when it goes higher than a defined threshold. RMSProp shows better performance for large size news articles as it does not suffer from vanishing gradient problem. The performance accuracy of the candidate model can be done using various available evaluation metrics. To check the prediction accuracy whether the news article is correctly classified as REAL or FAKE, accuracy as evaluation metric is used. Accuracy is calculated as ratio of number of correctly predicted samples to total number of samples for a given data set.

We've trained our simple LSTM model on a fake news dataset and got an accuracy of 99 %. There are many other machine learning models which perform much better but let's admit it Machine Learning models require a lot of feature engineering and data wrangling. We are using a deep learning model to let the model figure everything out on its own. We can further increase model size, and tweak the hyperparameters. We have also deployed an application on top of it.

Is it really learning what fake news is? No. If a good journalist were to write the same contents as the fake news article it would probably classify it as being real. Keep in mind that the model does not know the actual state of the world.The proposed model works well for the balanced high dimensional news data set. More thorough experiments will be required in the future to further understand how deep learning model with attention can help to evaluate the automatic credibility analysis of News.

V. Conclusions and future work

In and news articles to be read online. The growing problem of fake the 21st century, the majority of the tasks are done online. Newspapers who were earlier preferred as hard-copies are now being substituted by applications like Facebook, Twitter, news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards the use of digital technology. Lots of riots and mob lynching cases happened due to misinformation and fake news. We can solve this problem using Machine Learning. Fake news detection is proven using various Machine Learning and Deep Learning Techniques. Machine Learning Algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Neural Network Models are used to predetermine the future content and determine the inaccurate news and posts. Using these methods, we can get alerts about fake news and the user will get the correct information. This project is a small step towards stopping these misinformation and fake news.

The current project did not include domain knowledge related features, such as entity-relationships. Future studies could extract name entities from each pair of news headline and news body and analyze their relationships through a knowledge base. The study demonstrated that even the very basic algorithms on fields like AI and Machine Learning may find a decent outcome on such a critical issue as the spread of fake news issues worldwide. Accordingly, the aftereffects of this examination propose much more, that systems like this might come very much handy and be effectively used to handle this critical issue.

ACKNOWLEDGMENT

We would like to express our deepest appreciation and a special gratitude to our summer design project mentor, Dr. Bhupendra Kumar, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We thank all the people for their help directly and indirectly to complete our assignment.

REFERENCES

[1] Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H., 2017. Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), pp.22-36.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Abdullah-All-Tanvir, E. M. Mahir, S. Akhter and M. R. Huq, "Detecting Fake News using Machine Learning and Deep Learning Algorithms," 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, Malaysia, 2019, pp. 1-5, doi: 10.1109/ICSCC.2019.8843612.

[3] Olah, C., 2015. Understanding lstm networks.

Phi, M., 2018. Illustrated Guide to LSTM's and GRU's: A step by step explanation. Towards Data Science.