Mini Project Report on

FAKE NEWS DETECTION

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

For the subject of

Computational Lab – II (CSL804)

by

TANMAY TALELE (31)
MANISH PAWAR (32)
SANKET VAGAL (33)

Supervisor:

Dr. TATWADARSHI P. NAGARHALLI

Computer Engineering

Department VIVA Institute of

Technology University of

Mumbai

2019-2020

CERTIFICATE

This is to certify that the project entitled **"FAKE NEWS DETECTOR"** is a bonafide work of **"Tanmay Talele (31), Manish Pawar (32), Sanket Vagal (33)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering in Computer Engineering"**

Dr. Tatwadarshi P. Nagarhall

Mini-Project Dissertation Approval for B.E.

This project report entitled **FAKE NEWS DETECTOR** by **Tanmay Talele, Manish Pawar** and **Sanket Vagal** is approved for the degree of **Bachelor of Engineering in Computer Engineering**.

Examiners

1. ----------

2. ----------

Date:

Place:

Abstract

Fake news, deliberate disinformation, hoaxes, parodies and satire are various ways to mislead people in order to damage an agency, entity, or person, and/or gain financially or politically. Of late, fake news has been in the spotlight of mainstream journalism and the general public because of how it can have an effect on the political scenario of a country(Fake News). In this project, we attempt to detect authenticity of specifically political news. We work to improve upon the existing model of Fake news detection out there, inculcating various ideas we learned during the course of the subject.

Table of Contents

List of Figures

List of Table

Declaration

We declare that this written submission represents our ideas in my own words and where others ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will because for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Tanmay Talele

Manish Pawar

Sanket Vagal

Date:

Chapter 1

INTRODUCTION

Our main aim of the project is detection and classification of fake news. The primary channel for spreading such content is social media and it sometimes finds its way into the mainstream media as well. Day by day it is becoming increasingly important to detect and classify fake news as such, because of the grave impact it can have on the political results of an election. The primary
the challenge for solving the issue of fake news is how loose the definition of the term Fake news is. For e.g. fake news can be classified into various categories: a statement which is known to be completely false, or a speech stating some statistics as facts for which no real analysis has been done, or a piece of text which is satirical.

1.1 Problem Statement

The local ideas currently above fail to take into consideration the temporal and syntactic content of the statements. For e.g., Fake news sentences generally have a complex dependency parse as they tend to include a lot of phrases and punctuations. So, we have used POS tags of the sentences and the Dependency parsing of the sentence to try to include these information while training the models.

1.2 Description

In the baseline model, our bi-LSTM uses just the statement embeddings and metadata information. Now, we modify the architecture to include POS tags of the statements and the relations given by dependency parser. It removes human errors that commonly occur during manual checking. The system provides an unbiased result. We incorporate POS tags and dependency parse of the statement in our feature set which helps us learn richer representations of the data thus increasing the accuracy of predictions.

Chapter 2

LITERATURE SURVEY

This chapter gives brief information about the existing system and its analysis in form of an analysis table which includes features like main points to be drawn out with its pros and cons.

2.1 Literature review

Ray Oshikawa, Jing Qian, William Yang Wang et.al proposed survey on Natural Language Processing for Fake News Detection[3]. In this paper, they surveyed automated fake news detection from the perspective of NLP. Broadly speaking, they introduced the technical challenges in fake news detection and how researchers define different tasks and formulate MLsolutions to tackle this problem. They discuss the pros and cons, as well as the potential pitfalls and drawbacks of each task. More specifically, they provide an overview of research efforts for fake news detection and a systematic compari-son of their task definitions, datasets, model construction,and

performances. They also discuss a guideline for future research in this direction. This paper also includes some other aspects such as social engagement analysis like comprehensive review of Natural Language Processing solutions for automatic fake news detection, systematically analyze how fake news detection is aligned with existing NLP tasks, and discuss the assumptions and notable issues for different formulations of the problem and categorize and summarize available datasets, NLP approaches, and results, providing first-hand experiences and accessible introductions for new researchers interested in this problem.

Emily Alsentzer, John R. Murphy, Willie Boag et.al had a paper called Fake News Detection via NLP is Vulnerable to Adversarial Attacks[4]. Here, news plays a significant role in shaping people's beliefs and opinions. Fake news has always been a problem,which wasn't exposed to the mass public until the past election cycle for the 45th President of the United States. While quite a few detection methods have been proposed to combat fake news since 2015, they focus mainly on linguistic aspects of an article without any fact checking. In this paper, they argue that these models have the potential to misclassify fact-tampering fake news as well as under-written real news. Through experiments on Facebook, a state-of-the-art fake news detector, they show that fact tampering attacks can be effective. To address these weaknesses, they argue that fact checking should be adopted in conjunction with linguistic characteristics analysis, so as to truly separate fake news from real news. A crowdsourced knowledge graph is proposed as a straw man solution to collecting timely facts about news events. Models here can detect fake news only when they are under-written, for instance when the content is totally unrelated to the headline (so-called clickbait) or when the article includes words considered to be biased or inflammatory.

In Machine Learning Approach to Fake News Detection Using Knowledge Verification and Natural Language Processing[5] paper, authors Koya Kawashima Wenjun Bai Changqin of Kobe University, Japan explored "fake news" as a disinformation tool. They survey previous efforts in defining and automating the detection process of "fake news" and establish a new definition of "fake news" in terms of relative bias and factual accuracy. Then they devised a novel framework for fake news detection, based on their proposed definition and using a machine learning model. They proposed a hybrid framework for fake news detection, which repurposes the machine

learning model for incident classification. Incident classification model consists of 5 NLP features combined with 3 knowledge verification features in the form of questions related to the scope, the spread, and the reliability of the source. The ternary answers to these questions are obtained from the user submitting a textual incident report and can be verified independently by the system. The model includes 1 more feature related to sentiment analysis, and we use the automated readability index (ARI), which is a readability test for English texts.

2.2 Analysis Table

Table 2.1 Analysis Table

Title of paper	Summary	Advantages	Disadvantages	Accuracy
A Survey on Natural Language Processing for Fake News Detection	Outline promising research directions, including more fine-grained, detailed, fair, and practical detection models	meta-data and additional information can be utilized to improve the robustness and to suppress the noise of a single textual claim	relying too much on speakers' or publishers' information for judging may cause misclassification	NA
Fake News Detection via NLP is Vulnerable to Adversarial Attacks	In this paper, they argue that fact checking should be adopted in conjunction with linguistic characteristics analysis, so as to truly separate fake news from real new	Hybrid approaches combine the advantage of linguis-tic models and network models, which intuitively out-perform either of them.	-Atop the existing embeddings, more advanced model architectures are not experimented with.	1159/2236 correctly classified with 47% acc. and 2184/2721 correctly classified with 80% acc.
A Machine Learning Approach to Fake News Detection Using Knowledge Verification and Natural Language Processing	In this paper, authors define what constitutes fake news in the context of information warfare, and propose an automated method for fake news detection based on this definition.	most promising framework for fake news detection uses a combination of source and fact verification with a hybrid framework based on automating incident classification.	When the threshold is set a small number, clustering of news articles from various dimensions is hard to work	Probability of getting detected fake news was observed as 75-80%.

Chapter 3

METHODOLOGY

This chapter covers the framework, details of the design of the model, hardware and software requirements of the system as well as the methodology of the system.

3.1 Framework/model/working

The details of model and its description with hyperparameters and overall working is mentioned.

3.1.1 working

For each statement, we remove the stopwords and clip the statement to length of 15 words. If there are fewer than 15 words, we use post-padding to get a statement containing 15 words. As stated in the previous section, for each word in the statement, we use 100 dimensional word-vector representation obtained from GloVe. We used string matching to map similar identifiers together.

Example, ’television interview’ and ’tv interview’ fall under the same broad category and must be given the same label. For each statement (without removing the stopwords), we generated the POS tags for each word. For each statement (without removing stopwords), we considered dependency parses which are frequently appearing (punct, prep, pobj, compound, det, nsubj, ROOT, amod, dobj, aux) in the training set. For each statement, we pass it through the Embedding Layer with weights as the embedding matrix. This is further passed as input to the Bi-LSTM. Similarly, for each statement, we pass the POS tags through the Embedding Layer with weights as the POS embedding matrix. This is further passed as input to another Bi-LSTM. Similarly, we pass the DEP tags through the Embedding Layer with weights as the DEP embedding matrix. This is further passed as input to another Bi-LSTM. All the three Embedding Layers are non-trainable.

3.1.2 Model architecture

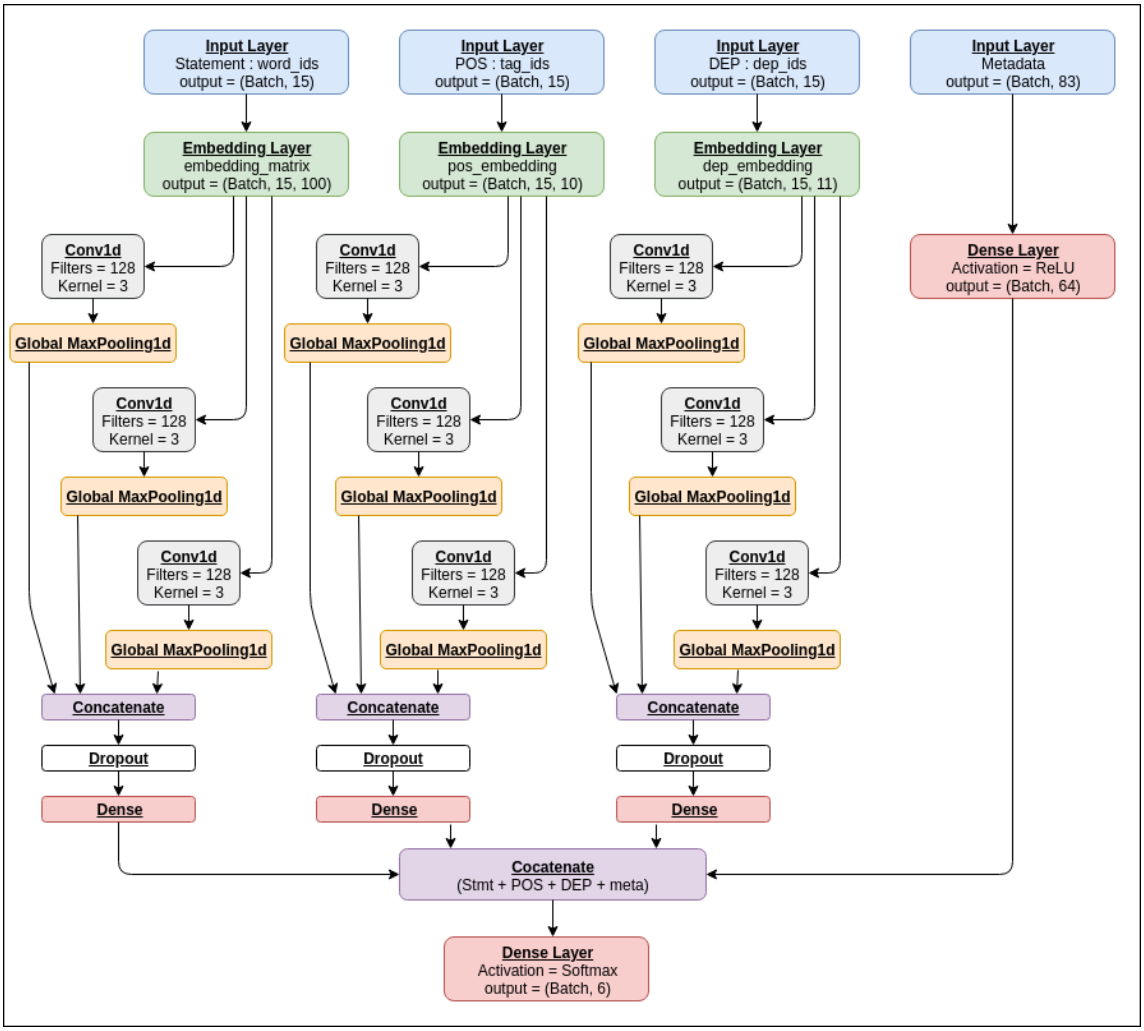

Fig.3.1 using tensorboard to generate LSTM model architecture

All the three bi-LSTMs have output layer size of 100 and a dropout of 0.2. The 83-dimensional binary feature vector of metadata is fed as input to a Dense Layer with ReLU as the activation function. We finally concatenate the outputs of these 4 above mentioned Layers and pass the output to a Dense Layer which serves as a classifier generating softmax probabilities over 6 output classes. This model is shown in Figure 3.1

3.1.3 Module Description

Dataset details:

LIAR dataset includes 12,836 manually labeled short statements from POLITIFACT.COM. It has annotated data for truthfulness, subject, context/venue, speaker, state, party, and prior history. With such volume and a time span of a decade, LIAR is an order of magnitude larger than the currently available resources. The fine-grained labels for the truthfulness ratings are : pants-fire, false, barely-true, half-true, mostly-true, and true.

Inputs:

For each training data, we clipped the statement to be of 15 words. Each of the 15 words would be represented by a 100-dimensional vector representation as from GloVe. Statement embeddings will have (None, 15, 100) shape.

Evaluation:

We used Accuracy as an evaluation measure. Since this dataset is a balanced one, it was observed that the accuracy results from different models were equivalent to respective measures. Hence, we used accuracy as the evaluation metric.

Evaluation measures:

The model will output the confidence value. The one with maximum confidence is chosen to be the prediction. This is compared to the ground truth values. Accuracy is chosen as an evaluation measure.

Hyperparam optimization:

Though Adam is computationally efficient and requires less tuning, for our case, we noted that the LSTM model overfits very easily with Adam. So, SGD with a learning rate of 0.025 and with nesterov momentum worked the best.

3.2 Requirements

3.2.1 Software needs

Python3.x - Python is one of the main languages being used in data analysis. As the data set was only for about 700 entries the packages available in python were sufficient for data processing.

Deep learning libraries - Keras, matplotlib==2.1.2, numpy==1.15.4, pandas==0.22.0 pydot==1.4.0s, scikit-learn==0.19.1, scipy==1.0.0, sklearn==0.0, spacy==2.0.18, tensorboard==1.9.0, tensorflow==1.9.0

SPACY - spacy==2.0.18

3.2.2 Hardware needs

Processors: AMD-RYZEN 3+ or Intel® CoreTM i7 processor.
Graphics: Nvidia GTX-960+ or RTX
Disk space: 100 MB.
Operating systems: Linux, macOS and Windows 8+
Python versions: 3.x

Chapter 4

IMPLEMENTATION and RESULTS

4.1 Code(files are attached with this report):

Dataprep.py -

This file contains all the pre processing functions needed to process all input documents and texts. First we read the train, test and validation data files then performed some preprocessing like tokenizing, stemming etc. There are some exploratory data analysis is performed like response variable distribution and data quality checks like null or missing values etc.

FeatureSelection.py -

In this file we have performed feature extraction and selection methods from sci-kit learn python libraries. For feature selection, we have used methods like simple bag-of-words and n-grams and then term frequency like tf-idf weighting. we have also used word2vec and POS tagging to

extract the features, though POS tagging and word2vec has not been used at this point in the project.

Classifier.py -

At this stage, we have 3 models viz. bi-LSTM, CNN, SVM that outputs the predictions. Each of these models are capable of learning representations which the other model may not be learning. We pass the input training sample to each of these three models and then train another MultiLayer Percep-tron over the predictions generated by these three models. Code will learn how much weight to give to each of the three models' predictions and then generate a final output label. However, the accuracy of this Hybrid model was not even close to that of the baseline model. This was not obvious at first, but could be attributed to one of the three models being a bottleneck in the training process. If good performance of one model can improve the performance of the overall hybrid model, the poor performance of the same model can bring down the performance of the overall hybrid model too.

Prediction.py

Our finally selected and best performing classifier was Logistic Regression which was then saved on disk with name final_model.sav. Once you close this repository, this model will be copied to the user's machine and will be used by prediction.py file to classify the fake news. It takes a news article as input from the user then model is used for final classification output that is shown to user along with probability of truth.

4.2 Results

Model	Features	Val Acc	Test Acc
bi-LSTM	stmt	26.09%	23.76%
bi-LSTM	pos	24.84%	22.57%
bi-LSTM	stmt + party	28.27%	24.86%
bi-LSTM	stmt + state	27.26%	25.26%
bi-LSTM	stmt + venue	27.18%	25.10%
bi-LSTM	stmt + job	26.64%	25.34%
bi-LSTM	stmt + subject	27.73%	26.20%
bi-LSTM	stmt + speaker	27.88%	25.34%

Fig.4.1 different models tested

We experimented with different models with different feature techniques. Our avg val accuracy was 26%.

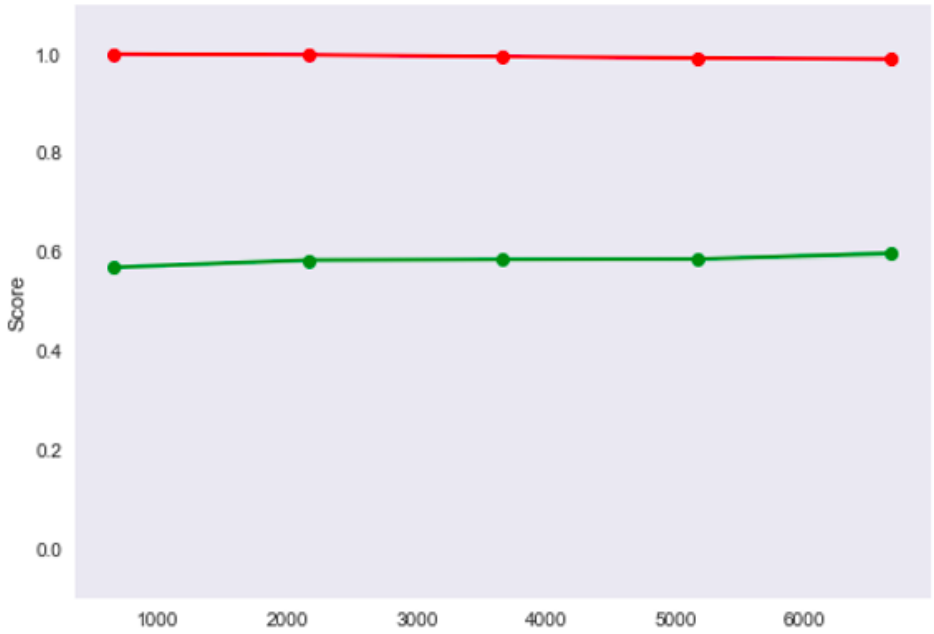

Fig.4.2 training and validation loss

Here, learning curve is shown where our probability score against training and validation epochs is scaled with loss in red color and accuracy in green color.

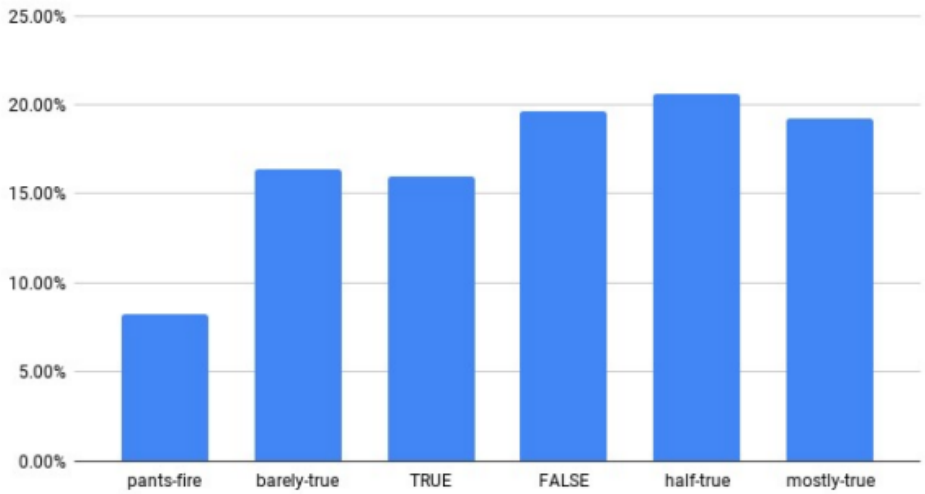

Fig. 4.3 Output label distribution

Here, the output labels in LIAR dataset was compared against the output label distribution probability.

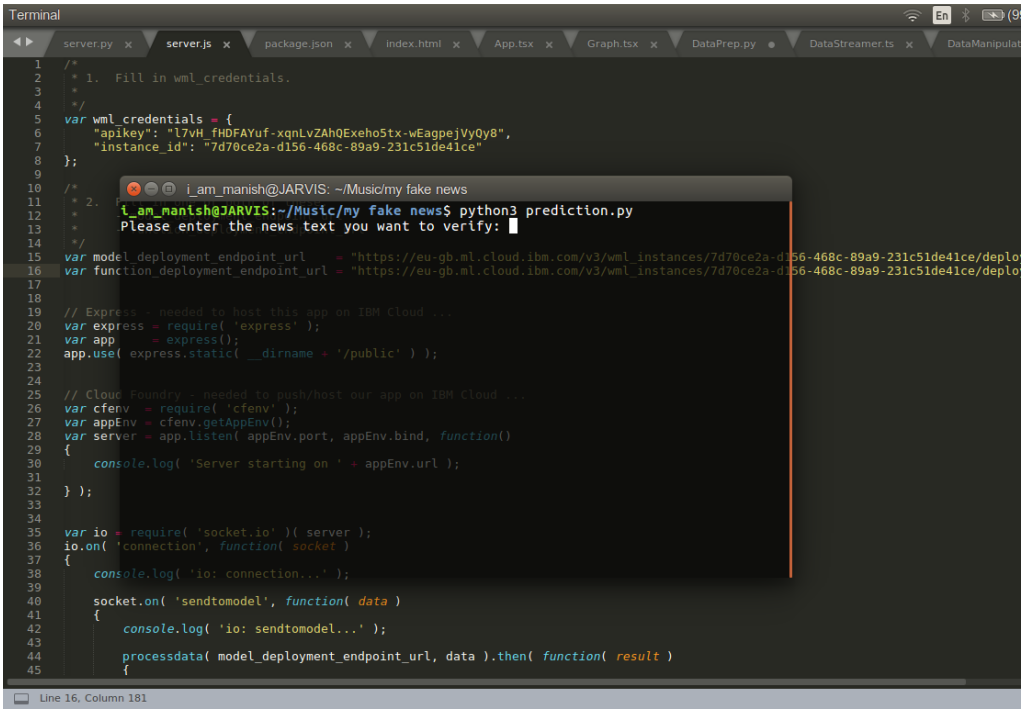

Fig. 4.4 Portal of Admin Activity to add students

It took approx 3.5 hours to train our model and when trained it was saved as .sav extension along with pickle .pkl and json as weights. The model path was specified to prediction.py.
Thus on executing prediction.py, it asks to enter the text we want to verify. We can put any text here and thus on inserting text here, it passes that to the trained model and the model outputs highest probability matched label from the output distribution.

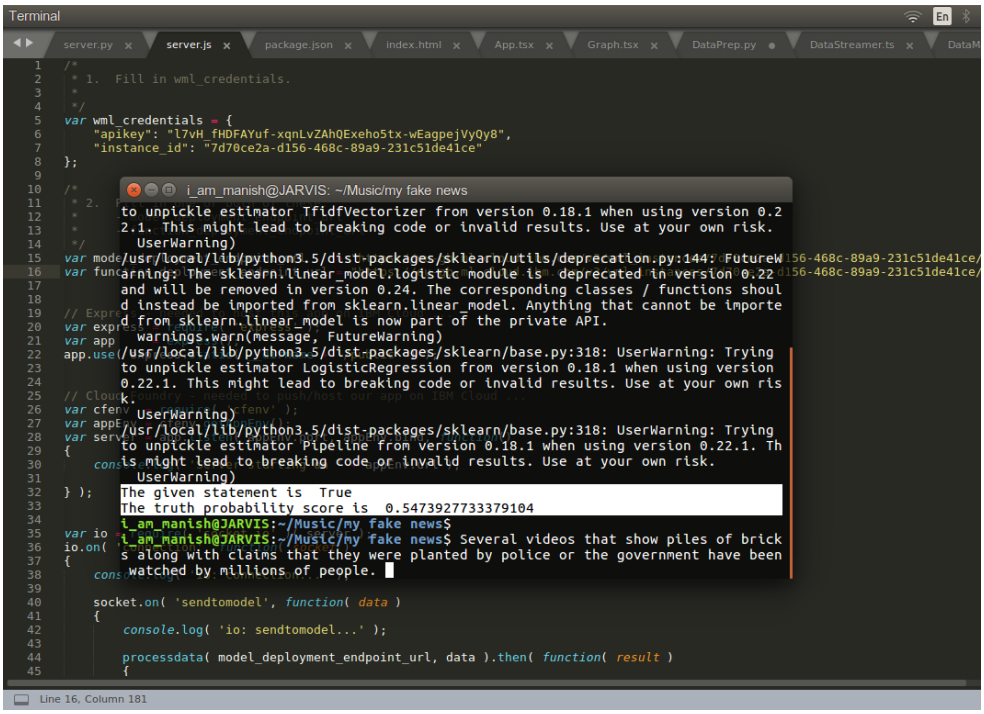

Fig. 4.5 After passing test news text

Thus, we can see that the sentence we inserted here is True news and not a fake one. The news was scraped from BBC[8] news webpage about recent american riot incidents.

Chapter 5

CONCLUSION

Conclusion

Our bi-LSTM model with rich input features (statement + metadata + POS + DEP) outperforms the best results of gpt-2 openAi model. POS tags and DEP parse information improves the performance of the baseline models. LSTMs are better than CNN for textual data. The take away from this project is that we understood what kind of problem requires which model architecture, based on the dataset spread and metadata.

Chapter 6

REFERENCE

References:

[1] https://www.arxiv.org/gpt2bert.html
[2] https://www.github.com/fakestance
[3] Ray Oshikawa, Jing Qian, William Yang Wang et.al “Survey on Natural Language Processing for Fake News Detection”,NIPS 2019,NY. https://arxiv.org/pdf/1811.00770.pdf
[4] Emily Alsentzer, John R. Murphy, Willie Boag et.al ”Fake News Detection via NLP is Vulnerable to Adversarial Attacks”, ICMP 2019, Seattle.
https://arxiv.org/pdf/1901.09657.pdf
[5] Koya Kawashima Wenjun Bai Changqin et.al “Machine Learning Approach to Fake News Detection Using Knowledge Verification and Natural Language Processing”, NIPS 2019, Japan
https://www.researchgate.net/publication/335191041
[6] https://www.bbc.com/news/world-us-canada-52941254