Fake News Detection

Arka Talukdar[1] Darshil Kapadia[2] Jay Virendra Pokarna[3] Vaibhav Jade[4]

CS F469 Information Retrieval Project

Birla Institute of Technology and Science, Pilani Campus

Pilani-333031

{f2015112[1], f2015113[2], f2015067[3], f2015115[4]} @pilani.bits-pilani.ac.in

ABSTRACT

A recent increase in abundance of "fake news" especially in social media and its negative impact on public decision making in democracies is a deep concern. Fake news is defined as "made-up stories written with the intention to deceive" and published in formats similar to those of traditional "real" news. We build a neural network based text classification system, trained on news articles published during the 2016 US Presidential Elections time period, to classify previously unseen articles as real or fake. Furthermore, we create a new dataset, having (both fake and real) Indian news articles, and attempt to fine tune the classification system to work well in the context of Indian news stories.

Keywords

Deep Learning; LSTM; Natural Language Processing; News; Fake News Detection

1. INTRODUCTION

While the spread of misinformation and biased (or outright false) news has been a problem since a long time, the capacity for doing the same has been increased exponentially due to advent of modern communication technology, especially social media. Numerous reports of fake news having influenced important elections (the US presidential election and the Brexit referendum for instance) justify this claim. Friends and acquaintances sharing a news story (thus adding a perceived stamp of legitimacy), the same type of news (based on the previous history of the user) being constantly presented to the user by automated recommender systems, reading the article partially along with personal prejudice, thoughts and opinions, can cause people to believe news stories which might be easily categorized as fake on careful, objective inspection. Thus, automated detection of possible fake news stories is a necessity in today's times.

Numerous methods have been employed for deception detection over the years. These methods can be broadly categorized into:

1. Linguistic Approaches in which the content of messages is extracted and analyzed to associate language patterns with deception
2. Network Approaches in which network information, such as message metadata or structured knowledge network queries can be harnessed to provide aggregate deception measures(Niall J. Conroy et al.,2015).

Our classification system is based on a Recurrent Neural Network (RNN) with Long Short-term memory (LSTM) units. Briefly, the classification pipeline works as follows: The headline and the body of the news article is first pre-processed to remove punctuation and unfold capitalization. The text is then vectorized (using two approaches: fixed pre-trained embeddings and dynamic word embeddings). The vectorized text is then fed into the recurrent neural network (RNN), having long short-term memory units (LSTMs), to get the classification output.

It should be noted that the RNN thus trained attempts to classify articles as fake or real based on the tone, word-usage, and the general structure of sentences found in the article. It does not fact-check the content of the articles. For instance, the RNN would not recognize an article titled as "Narendra Modi is the prime minister of Australia" as fake news if the tone of the article resembles that of a legitimate news story.

2. RELATED WORK

The problem of fake news detection is essentially a text classification problem. Although fake news is a fairly new domain, text classification has one of the most important tasks machine learning problems. A number of algorithms have been successfully implemented in this regard.
In recent times, Recurrent Neural Networks (RNNs) have shown good performance on text classification tasks. LSTM (Long Short-Term Memory) a special type of RNN have been particularly successful when used in tandem with word embeddings.
We apply this popular model for text classification on the Fake News Detection task. Work in Fake News detection has been approached in two ways, automated factual verification of news articles from trusted source and detection of fake news through the style and presentation of content.
We attempt the former in our paper identifying the styles and patterns inherent in fake news content using LSTM based neural networks and word embeddings.

3. DATA

We used two different datasets on which we are individually training and testing the neural network. The first dataset was a combination of news articles from Kaggle Fake News Dataset (containing about 13000) articles and mainstream US news agencies.The Kaggle fake This data has been obtained from the period of US 2016 elections. US news articles were collected using API provided by the vendors themselves.
We divided the data into 7:2:1 for training, validation and testing our model.
India is unique in its culture, geography, economy, politics. These things greatly influence the characteristics of news and are reflected from them. Hence the features learned from US dataset are not suitable for Indian news articles.
We used web crawling to scrape news articles from well-known news vendors in India. This includes credible vendors for real and humorous and satirical fake news websites. We have manually labeled this dataset according to the contents of the article.

Following is the number of articles per news vendors:

Indian Express	885
The Unreal times	49
Farzi news	103
News that matter not	87
Teekhi Mirchi	11
Hindustan Times	50

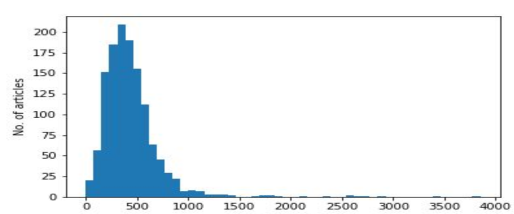
4. PROPOSED TECHNIQUE

The problem is formulated as a classification task and the objective is to learn a classifier using LSTM network. The proposed methodology involves a pipelined approach and is divided into four phases:

- Pre-processing
- Creating vector representations for the phrases
- LSTM model
- Training the model

4.1 Preprocessing

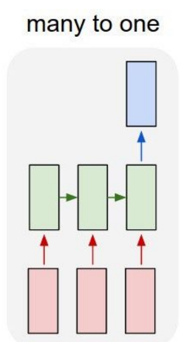
The datasets were pre-processed in order to ensure uniformity. Pre-processing included removal of special characters, numbers and converting all characters to lowercase. Also all numbers present in the dataset were removed and replaced by ‘<num>’ as the exact number is of no consequence here.
We found that most of the articles were about 500 words in length. We required our text input to be of fixed length so articles with word count greater than the limit were truncated and those below the limit were padded up.



4.2 Creating Vector Representation

LSTM based neural networks have been seen to give the best performance with dense word embeddings. We try two different word embeddings; pre-trained GloVe Vectors (trained on Wikipedia) and dynamically trained word vector on the training dataset of our news corpus.
For GloVe we use 50-dimensional word vectors. The word vectors make up the first layer of the LSTM model (embedding layer), for each word in the article a word vector is supplied by that article.

4.3 LSTM Model



Long-Short Term Memory units were used because the text is considered to be a continuous input as the words used earlier can affect words used later in the text. We used a many to one fixed LSTM, wherein we provide the network with an input sequence of fixed length and it produces a single output. Keras framework on top of Tensor Flow backend was used to build the model. The number of LSTM units in the model was 64 and a dense layer was added at the end to combine the outputs of the units to give a prediction.
We tried out different types activation functions (ReLU, Sigmoid) on the dense layer and optimiser functions (Adam, RMSprop) on our model.
We could have used larger network size and accompanying word embeddings could have been increased in dimension but our project was limited by the availability of hardware and time. We hope to try out larger and more complex network architectures in future

4.4 Training the Model

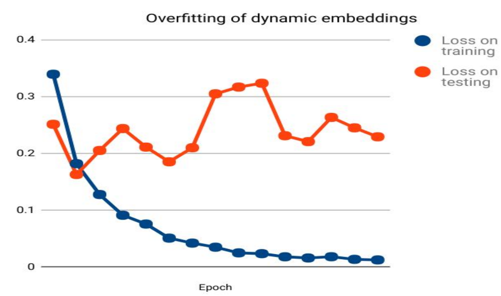
The network was trained for a total of 15 epoch with a batch size of 64. Initially, we trained the entire model on the US news dataset as it was significantly larger. We tuned the different hyperparameters on this dataset. We used the same hyperparameters on the Indian news dataset.
We evaluated Accuracy, Recall and Precision on the results

5. EXPERIMENTAL RESULTS

We had multiple choices on a number of hyperparameters and on each case we took decisions based on the performance of our classifier.

5.1 Word Embeddings

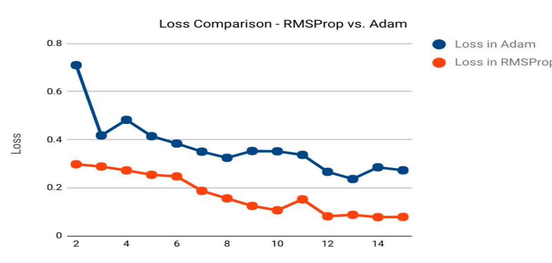
We had tried both pre-trained GloVe vectors and dynamic word embeddings. Both the word-embedding gave similar performance on training but when extended to validation set the performance dynamic word embeddings was poor. GloVe showed similar performance in both training and validation.



Hence it was a simple choice for us to proceed with GloVe word embeddings. We theorise that dynamic word embeddings failed to perform on testing data because it was overfit, while GloVe being trained on the entire Wikipedia corpus was much more resilient to new data.

5.1 Optimiser for Network

We initially tried Adam optimiser on our network as it has been seen to perform better in previous work on LSTM networks. However when we tried RMSprop optimiser our performance improved.



Thus we chose to use RMSprop for our model due to its better performance.

5.3 Activation Function

A number of activation functions have been tried on Neural networks with varying results. Due to time constraints, we could try out just two, 'Sigmoid' and 'ReLU'

Activation functions are very critical for a neural network as it is the source of non-linearity and determines the gradient during backpropagation. Activation functions can saturate gradient and give rise to vanishing gradient problem.

In our model, ReLU was outperformed by Sigmoid, which is unexpected as most of the literature indicates that ReLU to be a better choice.

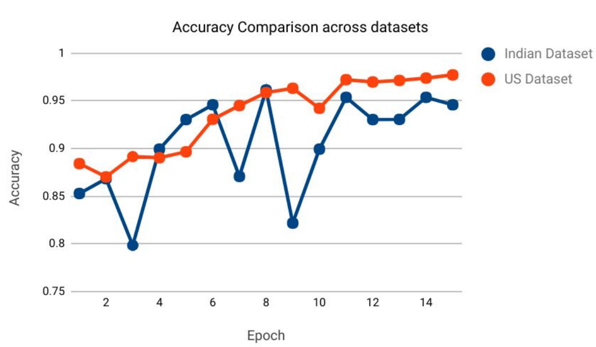
We would like to try out other activation functions like tanh and Leaky ReLU and investigate why our model shows better performance in Sigmoid.

5.3 Performance on Datasets

We implemented our fine-tuned network on both the datasets. As expected the US news dataset performed better and gave consistent improvements on training as it was much larger in size. The Indian news dataset compiled by our team showed slightly poorer performance and its performance fluctuated a lot while training.

We infer that the primary reason for this is the large difference in dataset size.

The graph below shows the accuracy difference between validation set while training.



5. EVALUATION RESULTS

On the US dataset, which was decent sized, the proposed method achieved an accuracy of 97.69% on previously unseen news articles. To ensure that the accuracy results didn't overestimate the effectiveness of the classifier, precision and recall were also calculated. Decent results, with precision of 98.62 and recall of 80.79 were obtained.

On the Indian Dataset, the accuracy, precision and recall were 94.57%,93.51% and 100% respectively. However, as shown in the graph above, the results were unstable and hence may not be indicative of the actual effectiveness of the classifier on Indian news stories.

We tried looking at the words that were the most commonly occurring in the fake news dataset (other than stop words):

They were (along with the percentage of documents they occurred in) :

india	51.0%	bjp	42.3%
modi	38.1%	comments	37.6%
delhi	35.4%	party	35.0%
minister	33.6%	rahul	33.4%
media	32.1%	congress	31.2%

This reflects that most of the fake news, like in US, revolves around politics. This poses a serious question as to whether our electorate is facing the same risk and is our democracy under threat of misinformation and polarization.

6. FUTURE WORK

There is a lot of scope for future work on this project. It includes trying out a larger and more elaborate network architecture and trying out different model hyperparameters. Additionally, in our present model, we do not include the article headline we can modify the architecture and include headline as a part of the input.

With respect to Indian context absence of API makes it harder for us to get articles but to get better results we need to broaden our sources and build a larger dataset.

Further work can be done on identifying the extent of bias and in news and it's direction rather than outright classifying it as fake and real.

7. ACKNOWLEDGEMENT

We would like to thank Dr. Poonam Goyal for giving us this opportunity, to complete this project under her supervision. We would also like to thank Ms. Chandramani Choudhary for mentoring us during the tenure of our project.

References

[1]] Yoshua Bengio and Patrice Simard and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks, 5, 157-166.

[2] Yoshua Bengio, Rejean Ducharme and Pascal Vincent. 2003. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155

[3] Pennington, J., Socher, R., and Manning, C., GloVe: Global Vectors for Word Representation, 2014.

[4] Hochreiter, S., and Schmidhuber, J., Long Short-Term Memory, Neural Computation, 1997

[5] Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classi- cation with recurrent and convolutional neural networks. arXiv preprint arXiv:1603.03827 (2016).

[6] Sapa Maheshwari. 2016. How Fake News Goes Viral: A Case Study. (November,2016). https://www.nytimes.com/2016/11/20/business/media/how-fake-news-spreads.html .

[7] Victoria L Rubin. 2017. Deception Detection and Rumor Debunking for Social Media. (2017).

[8] Victoria L Rubin, Yimin Chen, and Niall J Conroy. 2015. Deception detection for news: three types of fakes. Proceedings of the Association for Information Science and Technology 52, 1 (2015), 1–4

[9] Ilya Sutskever, Oriol Vinyals, and oc V Le. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems. 3104–3112.

[10] Lipton, Zachary C., Berkowitz, John, and Elkan, Charles. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019, 2015.

[11] Mikolov, T., Karafiat, M., Burget, L., ´ Cernoczk ˇ y, J. H., and Khu- ´ danpur, S., “Recurrent neural network based language model” Proc. of Interspeech 2010, pp. 1045–1048

Contributions

Vaibhav’s work was to clean and preprocess the US dataset and to create suitable vector representations for them.

Darshil’s work primarily concerned with the implementation of the classification pipeline. This involved converting the cleaned dataset into the suitable vector representation, implementing the LSTM-equipped RNN and training it with different hyperparameters (including the type of vector representation (fixed or dynamic), choice of optimizer and activation functions, general structure of the network,etc) in an attempt to enhance performance on unseen articles of the US dataset.

Jay’s work was related to web crawling. He scrapped articles and data from various Indian genuine and fake news service providers. He also preprocessed the dataset so that it can be directly used by the classifier.

Arka’s work was to fine tune the classifier and to implement it on the Indian News dataset. He has also done domain specific evaluation according to the news category. He built the GUI which has been provided along with the attached code. The report has been compiled by him.