Fake News Detection

Rajeev Bhatt Ambati
Electrical Engineering
rza67@psu.edu

Shaurya Rohatgi
Information Sciences and Technology
szr207@psu.edu

Neisarg Dave
Information Sciences and Technology
nud83@psu.edu

ABSTRACT

Given the human colossus that internet is in the day and age of social media, within no time it has the capability to make any post go viral. These conditions are the ideal platform for malicious entities to disseminate fake news. Social bots together with users who share posts without opening even exacerbates the situation. Fake news has an impact on the society and the decisions made by people. Recently there has been a lot of work done in this area of automatic fake news detection. Since the task of filtering the fake news is an arduous task for a human evaluator, there has been an increasing amount of research focusing on automatic detection of fake news in social media. Artificial Intelligence for Cyber Security (AICS) workshop under AAAI-18 hosted a mini-challenge in which the task is to propose techniques for the automatic detection of malicious news articles. We propose a technique which is simple and efficient at the same time to detect unreliable news and we train and report our results on the dataset provided by the challenge. We achieve better results than the baseline (Figure 1) mentioned and it paves a pathway for more experiments and approaches to be tried to further improve the performance.

KEYWORDS

fake new detection, classification, deep learning

ACM Reference Format:
Rajeev Bhatt Ambati, Shaurya Rohatgi, and Neisarg Dave. 2018. Fake News Detection. In *Proceedings of ACM Conference (IST 597: Principles of Artificial Intelligence).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Social media has become inseparable from our day to day activities since it is often more easy to consume news than the traditional media such as newspapers and television. But along with this comfort there is also a downside to it. We all know how fake news has played out in the past election cycle for the 45th President of United States. Fake news has also started to create real life fears: In 2016, a man convinced from an online news source that read "a pizzeria was harboring you children as sex slaves as part of child abuse ring led by Hillary Clinton" and carried an AR-15 rifle and walked in a Washington DC Pizzeria. He was later arrested for firing [Kang and Goldman 2016].

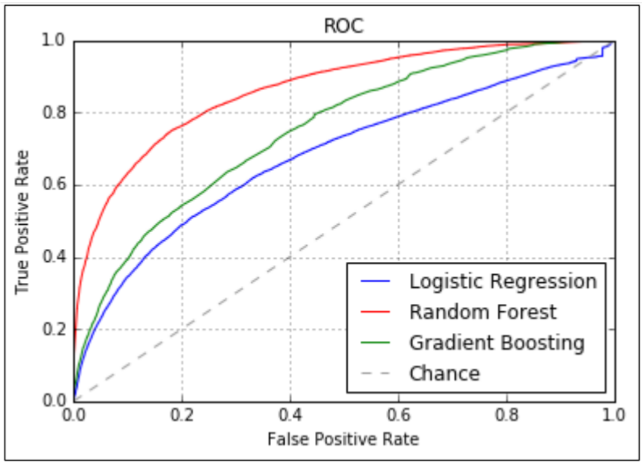

Figure 1: Challenge Baselines - AUC for - Logistic Regression-0.6808, Gradient Boosting-0.7526,Random Forest-0.8632

This extensive spread of fake news can have a serious negative impact on the individuals and the society. For example, during 2016 presidential election, the most authentic mainstream news was on a back foot compared to a most popular fake news circulated on Facebook [Buzzfeed 2016]. Fake news can tarnish the reputation of social media as a whole and thereby effecting authentic sources. Fake news is usually used by propagandists to convey or influence political decisions. For example, some reports show that Russia has used social bots to spread false stories by using fake accounts [TIME 2017]. Hence, its critical to develop methods to automatically detect fake news on social media.

2 PREVIOUS WORK

Though several attempts were made in the past to tackle fake news on social media, fake news detection is essentially a hard problem because there no a specific pattern in which fake news is disseminated. There are challenges that needs to be tackled in detecting the content. But solely focusing on content reduces the chances of detecting fake news. Fake news when spread by propagandists is highly organized as social bots are used. Hence a lot of information can be drawn from capturing the manner in which it is posted across forums. Though the quantity of fake news is more, it is spread from specific sources. Hence incorporating the source information of any kind also improves the likelihood of detecting fake news.

The use of news media for propaganda and disinformation operations is a serious threat to society. The technology to automatically and reliably detect unreliable news articles on the Internet does

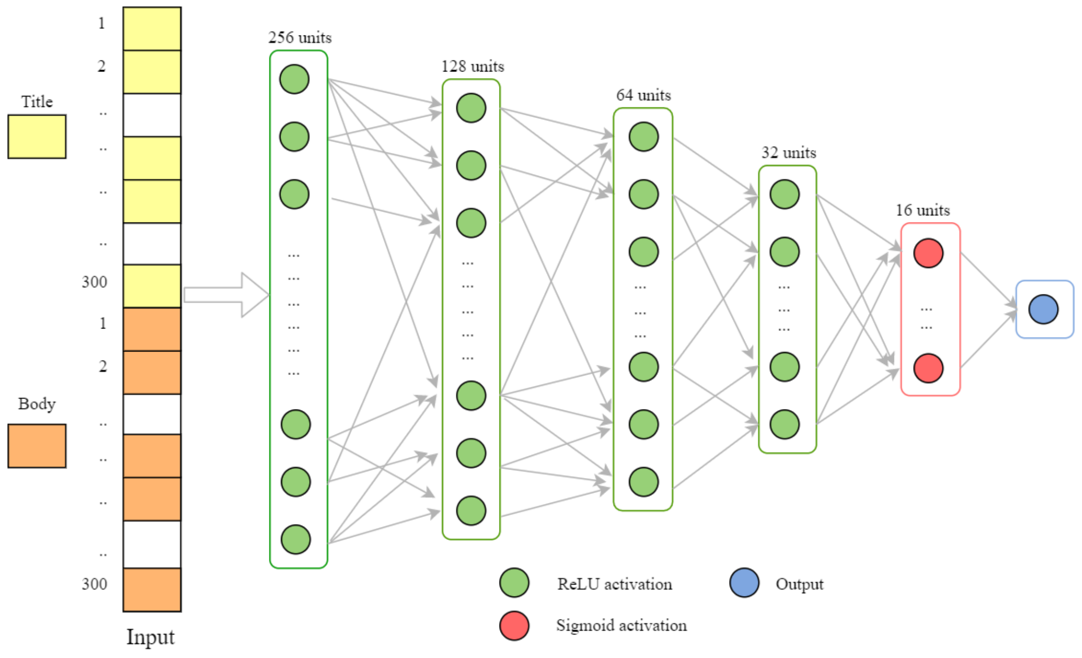

Figure 2: Architecture of our model, a 4-Layer MLP.

not currently exist. The goal of this challenge is to spur the development of novel automated methods to classify Internet news articles as unreliable or not. 1 At its core, this Challenge is a two-class discrimination problem. For the purpose of this Challenge, the categorization of an individual article as being reliable or unreliable is solely based on the designations found within Melissa Zimdars' (Professor of Communications, Merrimack College) Open Source Unreliable Media List (http://www.opensources.co/).

Unlike other machine learning problems, there are several ways in which features are extracted to perform fake news detection. Linguistic based features are extracted to capture the catchy news often called 'clickbait' in [FuÂÍrnkranz 1998; Yimin Chen and Rubin 2015] and deceptive writing styles in [Sadia Afroz and Greenstadt 2012]. Visual hand crafted and statistical features extracted from the articles are used in [Aditi Gupta and Joshi 2013; Zhiwei Jin and Tian 2017]. Some attempt has been made by [Carlos Castillo and Poblete 2011; Fan Yang and Yang 2012; Jing Ma and Wong 2015; Sejeong Kwon and Wang 2013] to capture the source information and incorporate wit with the content. After the feature extraction, [Naeemul Hassan and Tremayne [n. d.]; Vlachos and Riedel 14] uses a knowledge base for fact checking before feeding the article to further scrutiny. A convolutional neural network (CNN) is used by [Wang 2017] for capturing deceptive styles in the fake news.

Support Vector Machines (SVM) and other hand crafted features are used by [Conroy and Chen 2015; Pérez-Rosas et al. 2017; Riedel et al. 2017; Rubin 2016; Shu 2017]. All these approaches are either restricted by using these features or not capable of capturing complex structure often found in fake news. Though an attempt is made to using deep neural networks, we believe the capacity of deep learning is not used in its full potential. We augment the linguistic features that capture the style of writing with semantics of the articles. Distributed representations make this semantic representation of the news article possible. The best Area Under Curve (AUC) reported by the AICS challenge is 0.8632 by using a Random Forest Classifier.

In this paper, we explore a attempt made in order to increase the performance of fake news detection on the AICS dataset. The rest of the paper is organized as follows: Section 3 covers a few of the powerful word representations proposed so far, Section 4 discusses the types of neural networks developed and different procedures employed to train a neural network. Finally, Section 5 discusses thoroughly about the dataset used, neural network architecture and the results thus obtained are obtained in Section 6 and conclusion in Section 7.

3 WORD REPRESENTATIONS

In a dyadic interaction, language is a conspiracy between two people, one interlocutor utters sounds and the other interlocutor makes up meanings or representations. Therefore, in order for a machine to perform language tasks, learning word representations is a vital step. So far, many methods are proposed to learn word representations that perform very well in many language tasks such as document classification, question answering, named entity recognition and also information retrieval. [Mikolov et al. 2013a] introduced two log linear models: Continuous Bag-of-Words (CBOW) model and Continuous Skip-gram (SG) model. Both the model architectures are shown in Figure 3.

CBOW is a Neural Net Language Model (NNLM) with the non-linear hidden layer removed and the projection layer is shared for all words. In a context window, the training criterion is to correctly

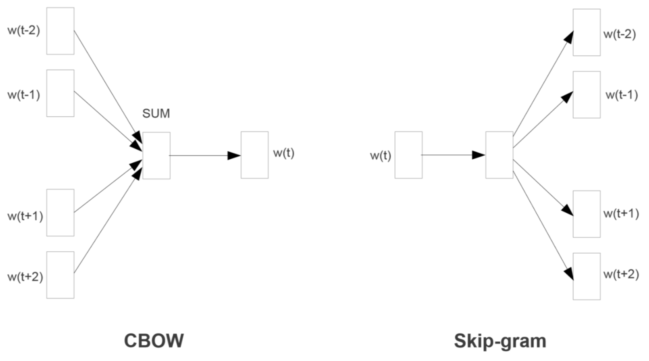

Figure 3: In CBOW, the middle word is classified given the context window. Whereas in Skip-Gram model, the other words in the context window are classified given the middle word.

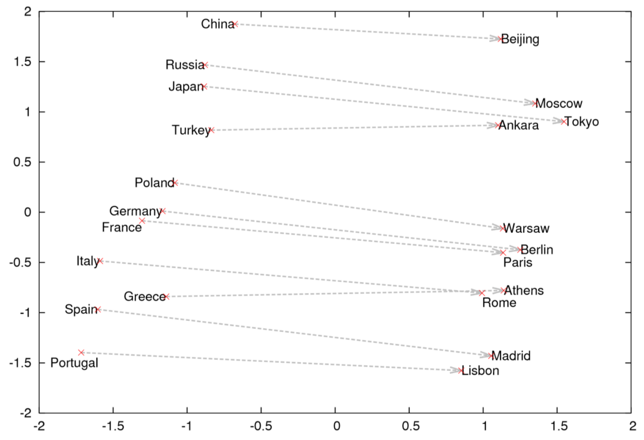

Figure 4: The first two principal components of the 1000 dimensional Skip-Gram vectors. The model is clearly able to organize concepts and learn the implications between words, here its country and respective capitals.

classify the middle word. The second model called as Skip-Gram model is similar to CBOW but instead of predicting the current word based on the context, it tries to maximize classification of a word based on another word in the same sentence. Using each current word as an input to a log-linear classifier, the objective is to predict every other word in the context window. Figure 4 shows the concepts learned by the Skip-Gram model without explicitly providing any information during the training. However, increasing the context window not only the increases the quality of the resulting word vectors, but also the computational complexity.

[Mikolov et al. 2013b] proposed different ways to improve the Skip-Gram model. First, the full softmax can be approximated using either a Hierarchical Softmax or Negative Sampling. Negative sampling basically reduces the tedious task of predicting every target word in the context window to distinguishing the target word and a draw from noise distribution using logistic regression. Here there are k negative samples, where k typically is small 2-5 in large datasets and thereby reducing the computational complexity. Second, it uses a subsampling approach since in a very large corpora, the most frequent words can easily occur hundreds of millions of times (e.g.,"in", "the" and "a"). Such words provide less information than the rare words and subsampling tries to counter the imbalance between them.

These word2vec models described above also hold vectors operations. Though they have succeeded in capturing fine-grained semantic and syntactic regularities using vector arithmetic, the origin of these regularities have remained opaque. [Pennington et al. 2014] proposes a model by combining the the advantages of global matrix factorization models and context windows based methods described above. It learns word embeddings that follows co-occurrence probabilities rather than probabilities themselves. This formulation of word vectors results in the following equation:

$$w_i^T + b_i + \tilde{b}_k = \log(X_{ik}) \tag{1}$$

Where, X is a matrix that tabulates the word-word co-occurrence count, w and $\tilde{w}$ are separate context word vectors, $b + i$ and $\tilde{b_k}$ are biases. It is trained by minimizing the cost function given below:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}))^2 \tag{2}$$

Where V is the size of the vocabulary. f is called a weighting function which has to satisfy required properties. Though a large number of functions satisfy, the the following class of functions are found to work well:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & x < x_{max} \\ 1 & otherwise \end{cases}$$

GloVe has shown promising results in many language tasks so far. For our project, we used this GloVe word vector representation that is pre-trained from a large corpora of sentences.

4 NEURAL NETWORKS

For many decades in machine learning research, it has been restricted to linear models like logistic regression, Support Vector Machines and other Kernel Machines. But with the advent of neural networks, we able to learn more powerful representations than ever. This is largely because of the universal approximation property of neural networks. Basically, a neural network learns powerful representations by hierarchically learning different levels of abstractions obtained from stacking many layers.

4.1 Architecture

The first layer into which the input is fed is called as the input layer and the one from which output is obtained is called output layer. All the rest of the intermediate layers present are called hidden layers. Each layer is basically computing many functions depending on the number of units that particular hidden layer has. These functions are basically affine transformations computed using different sets of weight vectors. There are other layers in between which apply non-linear functions like sigmoid, tanh and ReLU on top of this affine transformations, this is the basic source of non-linearity present in the neural networks.

Depending on the type of data that the neural networks handle, there are different kinds of neural networks. First, Feed Forward Neural Network, also called as a Multi-Layer Perceptron (MLP), which is just an extension of the Perceptron, one of the first neural network architectures. Briefly, it is a series of affine transformations coupled with an activation function. Since each hidden unit uses a different weight, usually millions of parameters has to be learned in a MLP. But most of the times, for examples while dealing with images, we don't need different weights. A Convolutioinal Neural Network (CNN) [Krizhevsky et al. 2012] is based on this idea and the functions that are constructed using different weights are called filters. Since there is parameter sharing in these networks, the number of parameters are reduced dramatically. These filters basically identify low level abstractions like horizontal and vertical edges in initial layers and high level abstractions like nose, lips, and ears in the deeper layers. A Recurrent Neural Network (RNN) is built to handle sequential data present in videos, stock market, and speech recognition. An RNN cell is instantiated multiple times and the input is fed into at different time instants through which the desired output is obtained. Typical RNN cells include Gated Recurrent Unit (GRU), Long short term Memory (LSTM) [Chung et al. 2014]. Both LSTMs and MLPs can be used for our problem but, since LSTMs operate sequentially, they take very long time to train on moderate GPUs. We used a Multi-Layer Perceptron for our project because it best suits our purpose.

4.2 Training

Now that we have established fundamental concepts about neural networks, the vital part is to train the neural network and thereby, learn the required parameters. The parameters of a neural network are found by searching the state space and setting them to the values that maximize our objective which is generally specified during the training process. So far an efficient procedure for searching the state space of neural networks has found to be the gradient descent optimization through *back propagation* algorithm. Gradient descent essentially updates the parameters by descending along the steepest direction and back propagation is a way to compute the gradients from the cost function.

Just as in any machine learning algorithm, neural networks are also prone to overfitting and since they are more powerful function approximators, they are even more capable of overfitting the data. Therefore, more sophisticated regularization techniques have to be used in order to optimize the performance of the neural networks on unseen data as well. [Srivastava et al. 2014] is one such technique that has shown a lot of success in the recent past. Dropout basically keeps a connection between neurons randomly for example the probability is sampled from a Gaussian distribution and thereby training an ensemble of networks. We have employed this dropout in our training procedure and turned out to be quite effective.

5 EXPERIMENTS

5.1 Dataset

For the purpose of the task to identify unreliable media and build automatic detectors an existing corpus of data containing news articles is chosen. These articles are then marked with labels signifying whether or not each article is reliable as determined by Open Source Unreliable Media List 1. From now on authors refer to Unreliable News Data 2017 (unr 2018) dataset as UND17. The dataset provided consists of predefined train and test splits and authors use the same splits for training and evaluation. Table 1 and 2 provide annotation counts and format of the dataset.

Table 1: The AICS'18 Challenge Dataset

	Reliable Articles	Unreliable articles
Train	95,295	34,524
Test	1,00,247	38,494

Table 2: Data Description

Field	Description
uid	Unique identifier for news article
title	Title of news article
text	Body text of news article
normalizedText	Cleansed body text of news article
label	Ground truth of news article

5.2 Semantic Representation of Title and Body

We first find the term frequency (tf) and inverse document frequency (idf) of every word from the text in train data. We use the distributed representations of words trained on Wikipedia corpus by Standford. They call them Glove embeddings. We believe that these representations capture the truthful nature of the words as they have been trained on Wikipedia corpus. We multiply the word embedding to its tf-idf weight. This helps us reduce the effect of more frequent words, which will have a notably lesser impact on the final representation of the text. Now we have words which are represented by their tf-idf weighted distributed representation. We go through every sample in our data and for every title and news article body we add these representations. At last we have a vector of 300 dimensions which represents the title and a vector of 300 dimensions which represents the body of the article.

We then concatenate these vectors and push them through our pyramidal multilayer perceptron model. The architecture of the model is discussed below.

5.3 Architecture

The features extracted from each article such as title and body are embedded using distributed representations of the word vectors. The resulting features of size 300×1 for each of the title and body are concatenated to form a 600×1 input vector. This is fed into a 4-layer MLP (Multi Layer Perceptron). The input layer has 256 units and each of the 4 hidden layers comprise of 128, 64, 32, 16 hidden units

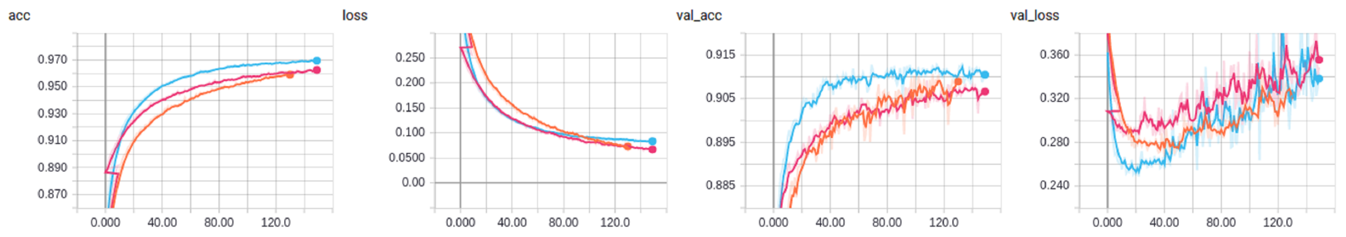

Figure 5: Varying layers in the model - Blue is with 4 layers, Pink with 3 layers and Orange is with 2 layers. We can clearly see as we go deeper the model converges quicker and the validation accuracy also remains higher throughout the 120 epochs.

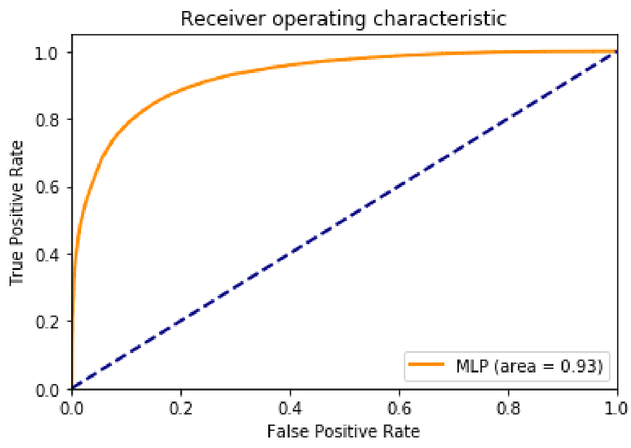

Figure 6: Best AUC for our MLP model after parameter tuning

respectively. The output layer has a single unit whose values lie between 0 and 1. Though it is a shallow network with fewer layers, using a ReLU (Rectified Linear Unit) activation function helps our model to converge faster. Hence we employed ReLU activation function before each of the 4 hidden layers. Finally, since our task is a binary classification problem, we have used a sigmoid activation function before the output layer. The combined architecture of our model is shown in Figure 1. The model takes about 1 hour to train. We used early stopping and dropout mechanism with a probability of 0.2 to avoid overfitting of the data. Our architecture is modeled by the following equations:

5.3.1 *Forward Propagation.*

$$\text{input layer}\begin{cases} z^{(1)} = W_1 X + b^{(1)} \\ a^{(1)} = ReLU(z^1) \\ a^{(1)} = a^{(1)} \odot I[I \leq p] \end{cases} \tag{3}$$

$$\text{hidden layers}\begin{cases} z^{(l)} = W_l a^{(l-1)} + b^{(l)} \\ a^{(l)} = ReLU(z^l) \\ a^{(l)} = a^{(l)} \odot I[I \leq p] \end{cases} \tag{4}$$

$$\text{output layer}\begin{cases} z^{(L)} = W_L a^{(L-1)} + b^{(L)} \\ y = \sigma(z^L) \end{cases} \tag{5}$$

$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & otherwise \end{cases} \tag{6}$$

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{7}$$

$$L = \frac{1}{N} \sum_{i=1}^{N} -\hat{y} \log(y) - (1 - \hat{y}) \log(1 - y) \tag{8}$$

Where ReLU and $\sigma(x)$ are the rectified linear unit and sigmoid activation functions respectively, $\odot$ denotes the element-wise multiplication or the Hadamard product operator. I is an indicator function samples from a Gaussian distribution of size $a^{(l)}$ and between 0 and 1 and p is the dropout probability. We have trained our model using the binary logistic regression loss shown in equation (8).

5.3.2 *Backward Propagation.*

$$\delta^{(L)} = \frac{dL}{dy} = \frac{1}{N} \sum_{i=1}^{N} -\frac{\hat{y}}{y} - \frac{1 - \hat{y}}{1 - y} \tag{9}$$

$$\delta^{(l-1)} = ((W^{(l)})^T \delta^{l+1}) \odot f'(z^{(l)}) \tag{10}$$

$$\frac{dL}{dW^{(l)}} = \delta^{(l+1)} (a^{(l)})^T \tag{11}$$

$$\frac{dL}{db^{(l)}} = \frac{1}{N} \sum_{i=1}^{N} \delta^{(l+1)} \tag{12}$$

$$\sigma'(z^{(l)}) = a^{(l)}(1 - a^{(l)}) \tag{13}$$

$$ReLU'(z^{(l)}) = \begin{cases} 1, & if\, a^{(l)} > 0 \\ 0, & otherwise \end{cases} \tag{14}$$

Where $\delta^{(l)}$ is the backward propagating error signal from the cost function at the output layer. This signal propagates and updates the derivatives as shown in equations (11) and (12). $f'(z^{(l)})$ is the derivative of the activation function used. In our case its the derivative of ReLU function in the hidden layers and sigmoid function at the output layer.

6 RESULTS

We built our model using Keras, a popular and sophisticated deeplearning framework and tensorflow is used as back-end. A GPU enabled tensorflow version is used and the model is run on a Nvidia GTX 1050 GPU which has 16GB of RAM and an i7 processor. We have also experimented with different layers in the MLP. 5 shows the change in performance versus the number of iterations. We can clearly see that the 4-layer model outperforms the other two models. We can also see that the training loss decreases while the val_loss decreases for a few iterations and thereafter starts increasing continuously. This is because the model doesn't converge and overfits the data if run for many iterations. To overcome this we have employed early stopping scheme and thereby running it for fewer iterations. Under these circumstances, dropout is known to help increase the model performance as discussed in section 4.1. Hence, we used a dropout with a probability of 0.2.

7 CONCLUSION

In this paper we present the effectiveness of the distributed word representations in this task. We evaluate our model for different parameter settings and report the results. We now have a thorough understanding of neural networks in terms of implementing, training and using typical regularization schemes like dropout and early stopping as when required to make them work. Since there is a sequential pattern in language data, LSTMs are known to help in these tasks. Since Convolutional neural networks are prominent in document classification, we believe 3D-Convolutional Neural Networks can have an advantage in capturing sequential pattern also. That would be the follow up if enough time and computational resources are available. The results obtained are promising and encourages for further exploration of novel methods. The authors would like to thank AICS 2018 Organizing Committee for timely support.

REFERENCES

Ponnurangam Kumaraguru Aditi Gupta, Hemank Lamba and Anupam Joshi. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. *In WWW* (2013).

Buzzfeed. 2016. (2016). https://www.buzzfeed.com/craigsilverman/viralfake-election-news-outperformed-real-news-onfacebook?utmterm=.nrg0WA1VP0#.gjJyKapW5y

Marcelo Mendoza Carlos Castillo and Barbara Poblete. 2011. Information credibility on twitter. *In WWW* (2011).

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

Victoria L. Rubin Conroy, Niall J. and Yimin Chen. 2015. "Automatic deception detection: Methods for finding fake news.". *Proceedings of the Association for Information Science and Technology* (2015), 1–4.

Xiaohui Yu Fan Yang, Yang Liu and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (ACM '17)*. ACM, 3.

Johannes FuÂĺrnkranz. 1998. A study using n-gram features for text categorization. *Austrian Research Institute for ArtiïňĄcal Intelligence* 3 (1998), 1âĂŞ10.

Zhongyu Wei Yueming Lu Jing Ma, Wei Gao and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. *In CIKM* (2015).

Cecilia Kang and Adam Goldman. 2016. In washington pizzeria attack, fake news brought real guns. *In the NewYork Times* (2016).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781 http://arxiv.org/abs/1301.3781

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *CoRR* abs/1310.4546 (2013). arXiv:1310.4546 http://arxiv.org/abs/1310.4546

Chengkai Li Naeemul Hassan and Mark Tremayne. [n. d.]. ([n. d.]).

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic Detection of Fake News. *CoRR* abs/1708.07104 (2017). arXiv:1708.07104 http://arxiv.org/abs/1708.07104

Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. *CoRR* abs/1707.03264 (2017). arXiv:1707.03264 http://arxiv.org/abs/1707.03264

Victoria L Rubin. 2016. "Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News.". *Proceedings of NAACL-HLT* (2016).

Michael Brennan Sadia Afroz and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. *In ISSP* (2012).

Kyomin Jung Wei Chen Sejeong Kwon, Meeyoung Cha and Yajun Wang. 2013. Automatic detection of rumor on sina weibo. In *Prominent features of rumor propagation in online social media (In ICDMâĂŹ13)*. IEEE, 1103âĂŞ1108.

Kai Shu. 2017. "Fake News Detection on Social Media: A Data Mining Perspective.". *ACM SIGKDD Explorations Newsletter* 19.1 (2017), 22–3.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

TIME. 2017. (2017). http://time.com/4783932/inside-russia-social-media-waramerica/

Andreas Vlachos and Sebastian Riedel. 14. Fact checking: Task deïňĄnition and dataset construction. *ACL* (14).

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *CoRR* abs/1705.00648 (2017). arXiv:1705.00648 http://arxiv.org/abs/1705.00648

Niall J Conroy Yimin Chen and Victoria L Rubin. 2015. Extracting and Composing Robust Features with Denoising Autoencoders. In *In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (ACM '15)*. ACM, 15–19.

Yongdong Zhang Jianshe Zhou Zhiwei Jin, Juan Cao and Qi Tian. 2017. Novel visual and statistical image features for microblogs news veriïňĄcation. In *IEEE Transactions on Multimedia (IEEE '17)*. IEEE, 598–608.

Received December 2017