Fake News Detection: Sequence Models Annual Bio-Inspired Conference 2018

Lin Peng
u6071322

Australian National University
ANU Research School of Engineering & Computer Science
Canberra, ACT 0200

Abstract. Sequence models such as Recurrent Neural Network (RNN) are most effective as processing textual information, especially with different input and output lengths. As an extension to the pilot research conducted on a 2-layer Neural Network with Bimodal Distribition Removal (BDR) algorithm, this current research continues the investigation on achieving a better model performance and examines possible solutions in solving the data sparisty issues using sequence models and denser word vector representations. Thus, the purpose of this current research is to investigate and compare the results of using a RNN with and without BDR using Word2Vec embeddings. The results of using RNN with BDR using Word2Vec and *tanh* achieve up to 80% model accuracy and faster training. While model accuracy without BDR achieve a stablized performance of 70%. As a result, the current research does show an improved model where the issues of overfitting is no longer present given by the Word2Vec technique in comparision to the pilot study.

Keywords: Deep Learning, Bimodal Distribution Removal, Outlier Removal, Fake News Challenge, Backpropagation, Forward Feed Neural Nework, Word Embeddings, Word2Vec, Recurrent Neural Networks, Sequence Models

1 Introduction

Fake news and the proliferation of raw opinion that passes for news create confusion about the basic facts of current issues and events [1], [2]. More importantly, the spread of fake news can impose potential impact on both politics and innocent individuals [1], [2]. Although, many Americans do believe fake news is sowing confusion whom were surveyed by the Pew Research Center, 39% Americans expressed high confidence in being able to recognize fabricated news; while another 45% feel somewhat confident [2], [10]. Yet, 32% Americans overall recognized political news stories online are often made up [2], [10]. Given that it is difficult and unrealistic to identify stories that may have been fabricated and potential hoaxes online based on traditional human-based fact checkers, technologies and techniques such as deep learning (DP) and natural language processing (NLP) hold promise for significantly automating parts of the procedure human fact checkers use today [3], [11]. A resonable approach for fact checking is the ability to detemine the stance of each document with respect to the claim [12]. In other words, the automation of this particular fact checking process is called Stance Detection, which can be leveraged by using DP and NLP techniques [3].

The task for the DP model is to find a function that can estimate the relative relationship between two pieces of text, which are the stance of the news body relative to a given headline [3]. Given that Stance Detection involves estimating the relative stance of two pieces of text relative to a topic, claim or issue, the key in the task of stance detection is to find out good features to represent the relations between headline and body text towards a given target [12], [13]. As an extension of the pilot research where a 2-layer neural network trained with Biomodal Distribution Removal (BDR) algorithm, the current research further investigate a different neural network architecture alongside with BDR and a different word embedding technique [14]. The pilot research has shown a skewed result outcome from the model because of data sparsity after textual to numerical feature extraction [14]. In other words, the original model of the pilot research resulted in an overfitted model despite the use of BDR [14]. One of the major reasons that data sparisty occured was due to the chosen word embedding technique, term frequency-inverse document frequency weighting (TF-IDF) [14].

To address the concern, in the current research, we develop a sequential neural network architecture with the use of Word2Vec word embedding techinque. In detail, a multi-layer recurrent neural network model (RNN) is implemented to further investigate the efficiency on the performance of the same datasets from pilot study. In addition, past research on the investigation of stance detection using FNC-1 datasets with deep learning sequential models such as RNN and LSTM have proven an improved results in comparison to a forwardfeed backpropagation neural network. According to Abeywardana, the RNN model using TF-IDF achieved an estimate of 87% accuracy on a set of news articles written around the 2016 U.S. election period [4]. On the other hand, research has also shown that given the datasets by FNC-1 of training 26,970 words soon became computationally unfeasible to train on a RNN model [15]. As a result, we have hypothesized that using a denser word embedding techinque trained on RNN alongside with BDR would produce significally better model accuracy in comparison to a 2-layer forwardfeed backpropagation neural network with BDR.

1.1 Sequence Models

Recurrent neural networks are a family of neural networks within the DP architecture that specializes in processing sequential data [26]. RNNs are most effective at tasks that involve sequential inputs, such as speech and language [25].

The data inputs for this current research are individual document as referred to as news articles consisting of the headline and the body content of the article. The output is a label classifying if the headline and body content of each input agrees, disagrees, discusses, or unrelates to one another. Although, the output length will be the same throughout the entire training, each input length will differ given that each headline plus the body content together will have a different length for each news article. Therefore, sequence models such as RNN is an applicable architecture to use in this research problem [26]. Moreover, similar research have been conducted in investigating the use of RNN in combating fake news [16].

2 Method

In this research, the method implemented consists of multiple properties including the RNN topology, Word2Vec, cross entropy loss function, activation functions, BDR and evaluation measures. The problem of the task is to classify the correct target label based upon the attributes from the dataset similar to the common objective for a multi-class classification problem. The training and testing corpus consists of four target labels classified as 'agree', 'disagree', 'unreleated', and 'discuss'. The two main attributes are the headline and body content of an article. In addition, it is a supervised multi-class classification because the training corpus have the desired target labels from FNC-1 [1]. In other words, the training of the network is to minimize the negative log probability of the correct output [5].

The approach consisted of preprocessing the datasets including the training and testing corpus provided by FNC-1 [1]. After preprocessing the datasets, Word2Vec technique is applied to represent dense vector representations of words from each input. With a denser vector representation of each word, a 3 hidden layer Elman RNN is implemented alongside with different hyperparameters. The activiation function *tanh* was used for the sequence model. Furthmore, the cross-entropy loss function was used to measure the performance of this classification model. The total number of epoch is 500 based on default conventions, and the learning rate is set to 0.01 after experimentation with 0.01 and 0.001, in which 0.01 delivered a better result.

2.1 Data Set

The training and testing data sets are provided by FNC-1 [1], which are derived from Emergent, consisting of a noval dataset from a digital journalism project for rumour debunking [17]. There was a total of 1683 document bodies, 49972 distinct headline as shown a smaple in table 1 and 2. And multiple headlines had the same body id.

Table 1. Sample Training Bodies Set

Body ID	Article Body
0	A small meteorite crashed into a wooded area i...
4	Last week we hinted at what was to come as Ebola fears spread across America..
5	(NEWSER) Wonder how long a Quarter Pounder with cheese can last? Two Australians..

Table 2. Sample Training Stances Set

Headline	Body ID	Stance
Police find mass graves with at least '15 bodies'...	712	unrelated
Hundreds of Palestinians flee floods...	158	agree
"Christian Bale passes on role of Steve Jobs..."	137	unrelated

Thus, the training corpus for this research was joined by body id, consisting of the body id, article body, headline, and stance. In other words, each input data from the training and the testing was the joined together by the body id resulting in headline and body of the article as shown in table 3. As a result, there was a total of 49972 data inputs for the training corpus. There was a total of 904 document bodies and 25413 distinct headline for the testing dataset. After joining each distinct headline with its corresponding body, there was a total of 25413 testing set. Due to the large amount data, the current model is trained on 5000 randomly selected data inputs and further split by 80% into training corpus. In addition, the four target variables were - 'agree', 'disagree', 'unrelated', or 'discuss'.

Preprocessing Each article from both the training and testing corpus were preprocessed using the NLKT tokenizer. Common English stopwords, digits, and URLs were removed. However, stemming and lemmatization were not being implemented within this research, partly because other studies on the same problem using different implementation of neural networks have shown that stemming and lemmatization did not have much impact on the overall model accuracy and result [20], [21].
The textual preprocessing from text to numbers are listed as the following steps:

Table 3. Sample Training Corpus

Body ID	Article Body	Headline	Stance
0	A small meteorite crashed into...	Soldier shot, Parliament locked...	unrelated
0	A small meteorite crashed into...	Tourist dubbed Spider Man...	unrelated
0	A small meteorite crashed into...	Luke Somers 'killed...	unrelated

1. Tokenization
2. Retain abbreivations
3. Remove capitalization
4. Remove digits
5. Remove URLs
6. Convert every letter to lowercase

2.2 Vector Representations of Words

Word2Vec algorithm was chosen direclty from the future work of the pilot study [14]. Word2Vec is the numerical representations of contextual similiarities between words combined with Continuous bag of words (CBOW) and Skip-gram model as shown in figure 9. As opposed to TD-IDF which is frequency or count-based, Word2Vec is predictive or probability.

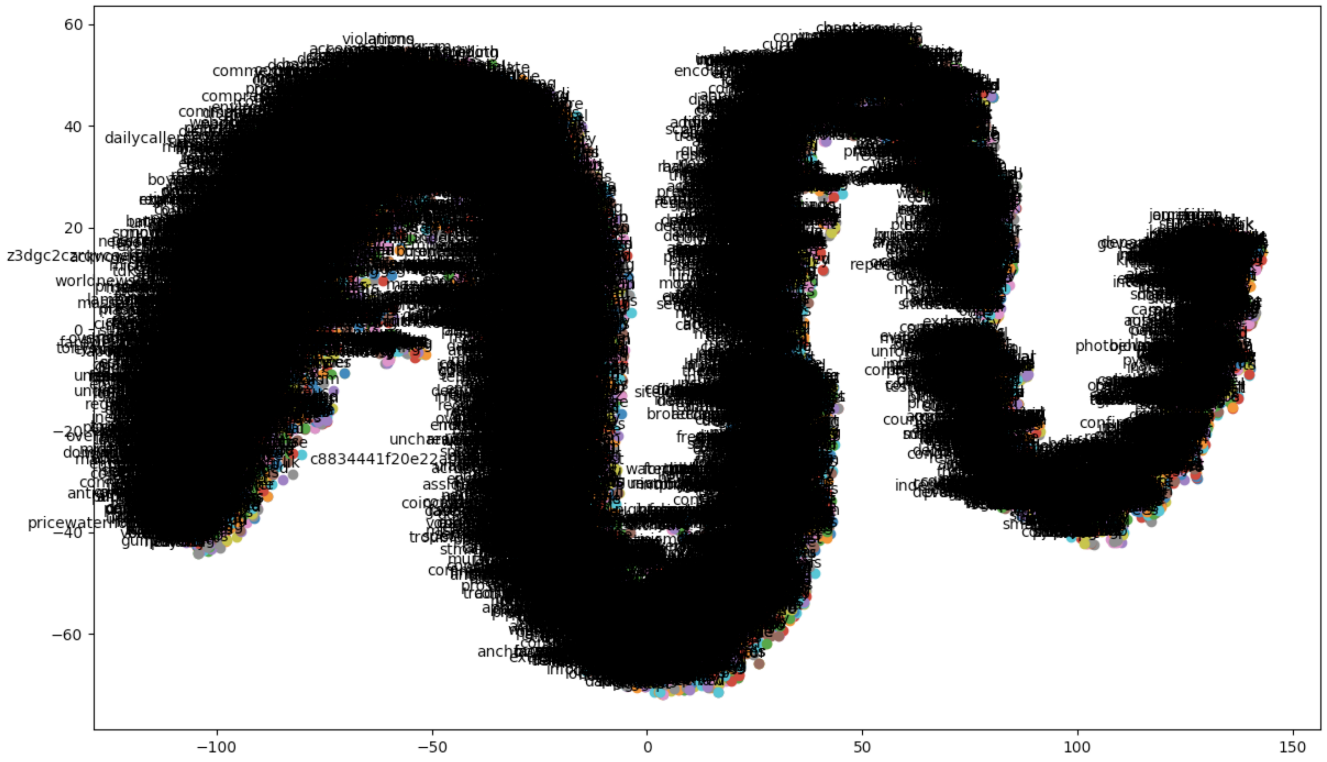

Fig. 1. This t-SNE visualization has a total of 19300 words from the vocabulary which was gathered and trained from the 5000 inputs of the training corpus

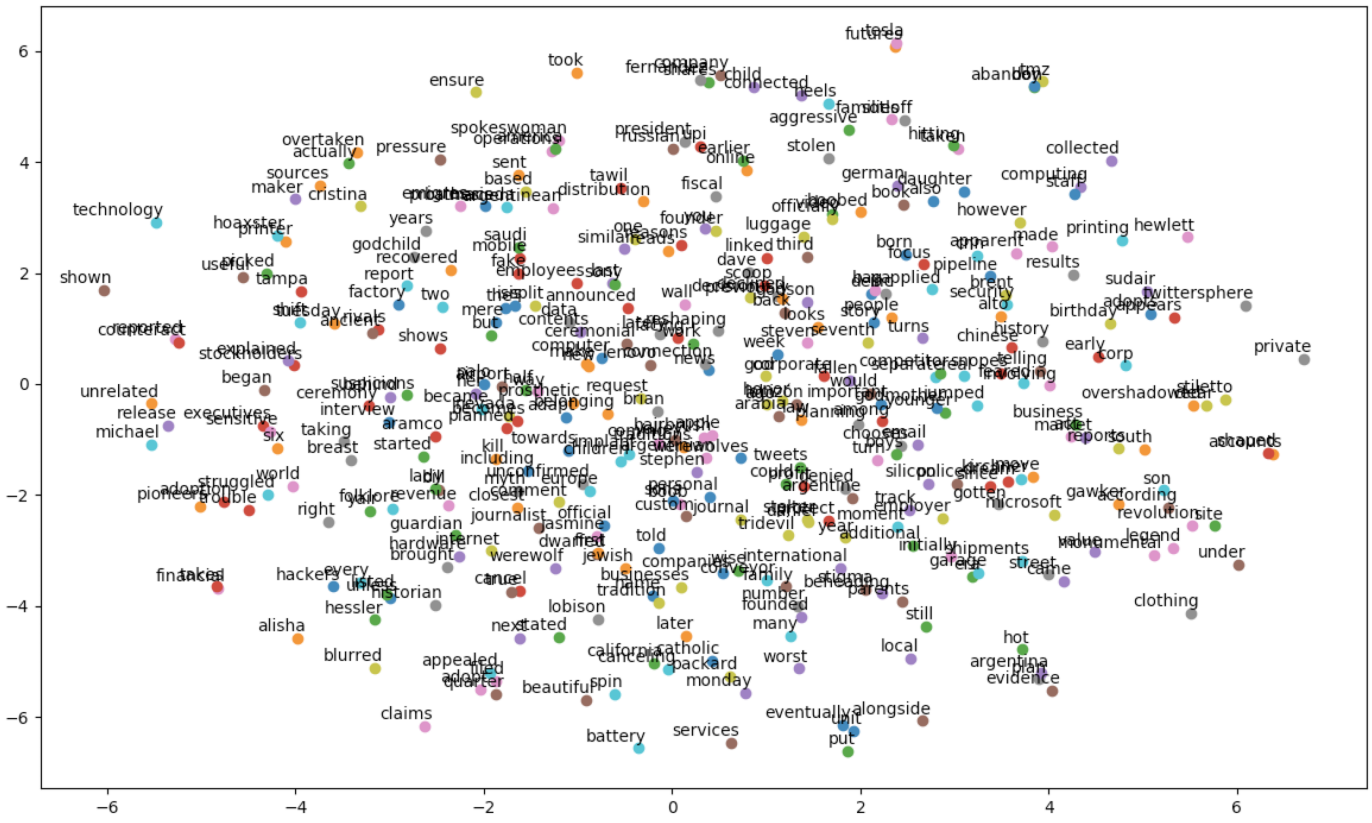

Fig. 2. This is a smaller test case of only 5 inputs of the testing corpus with a vocabulary size of 869 to show a sample display of Word2Vec similiarities. As shown 'Tesla' and 'Future' were similiarily grouped together thus their word embeddings are very much alike with similar features and scores.

The dimension specified for each word vector is 1x100 following the default conventions [23]. The context window specified is 1 to reduce the variance among word similiaries. Input word is represented by a one-hot vector multiplying to the cosine similarity within the context window which gives the embedding vector for the context word. Then it is fed into a softmax model:

$$\frac{e^{\theta_t^T e_c}}{\sum_V^{j=1} e^{\theta_j^T e_c}}$$

$\theta_\mathbf{t}$ is the parameter associated with output of a particular word. **c** is the context word. **V** is the vocabulary size. The output vector of each word is the input vector of the word but trained and fed through the softmax model.

Given that one of the major weaknesses of the previous research was data sparsity in using TD-IDF which resulted in the inability to call the Bimodal Distribution Removal (BDR) algorithm; Word2Vec is able to produce denser vectors for each word.

2.3 Bimodal Distribution Removal

As a continuation of the pilot study, BDR is also implemented within the current study. BDR is used to remove errors such as outliers that contributes to the bias-variance dilemma [24]. Thus, BDR is a technique within the training to examine the behavior of outliers. During training, the errors result in fluctuating variances from the mean. In the current study, the following steps were carried:

Algorithm 1 Bimodal Distribution Removal Algorithm

```
for epoch in total epoch do
    Calculate the normalized variance_ts of the normalized pattern errors
    if variance_ts < 0.1: then
        Calculate the normalized mean_error_ts of the pattern errors
        if each pattern error > mean_error_ts:
        Store the patterns
        Calculate the mean_error_ss of the stored patterns
        Calculate the standard_standard_deviation_ss of the stored patterns
        if each pattern with error ≥ mean_error_ss + α standard_deviation_ss:
    else
        halt if variance_ts <= 0.01
    end if
end for
```

During BDR, pattern error past the threshold the subset mean + α subset standard deviation will be removed. Those removed pattern errors are considered as outliers of the entire training corpus [24]. Furthermore, the training corpus will be trained dynamically as the pattern errors are removed, thus reducing the number of large datasets.

2.4 Recurrent Neural Network Model

An Elman recurrent neural network is used in the current work. It is based on a feedforward neural network with additional context neurons, which receive input from the hidden layer neurons. Different implementations of the Elman model was tested. The most basic Elman model consists of only one hidden layer where Many-to-One mapping was applied. Many-to-One mapping is where a sequence of inputs consist of many words i.e. 'residents reported hearing a loud boom Saturday night' and the output of the hidden unit from the first hidden layer will be a classified predicted label. The Many-to-One mapping can also be referred to as a RNN classification model because we would like to feed in each word within a sequence of input at each time step and at the last time step output the predicted label [6], [19]. For each word in the input sequence, the first hidden layer computes the following function:

$$h_t = tanh(w_{ih}x_t + b_{ih} + w_{hh}h_{t-1} + b_{hh})$$

$\mathbf{h_{t-1}}$ is the hidden state at previous layer t. The time is represented by each of the word in the input sequence. For example, in figure 1, each time step is represented by each word in the RNN. And from each time step, the parameters including the weight matrices $\mathbf{w_{ih}}$ and bias b_{ih} are learned using supervised training procedure, backpropagation through time. The output would be a sequence of the word vector and the last contextual embedding $\mathbf{h_t}$ represents the context per a news article. In other words, the parameters in RNN are pass through each time step and learned for the next given state including the hidden and activation units [26].

Activation Function The most common *tanh* function for sequence models was tested. And *tanh* activation function was used as opposed to using *sigmoid* or *Relu* because *tanh* has the ability to keep the gradient within the linear region of the activation function and can minimize the vanishing gradient problem given that it outputs values between -1 and 1 thus it will not result in a quick convergence to 0 as compared to *sigmoid* where its values are between 0 and 1; while *ReLu* can only fire if the output is above 0. In other words, *ReLu* computes the function of the max of 0 and the input. To prevent training from dying or result in no activation of neurons, *tanh* was the approprate choice over *ReLu*.

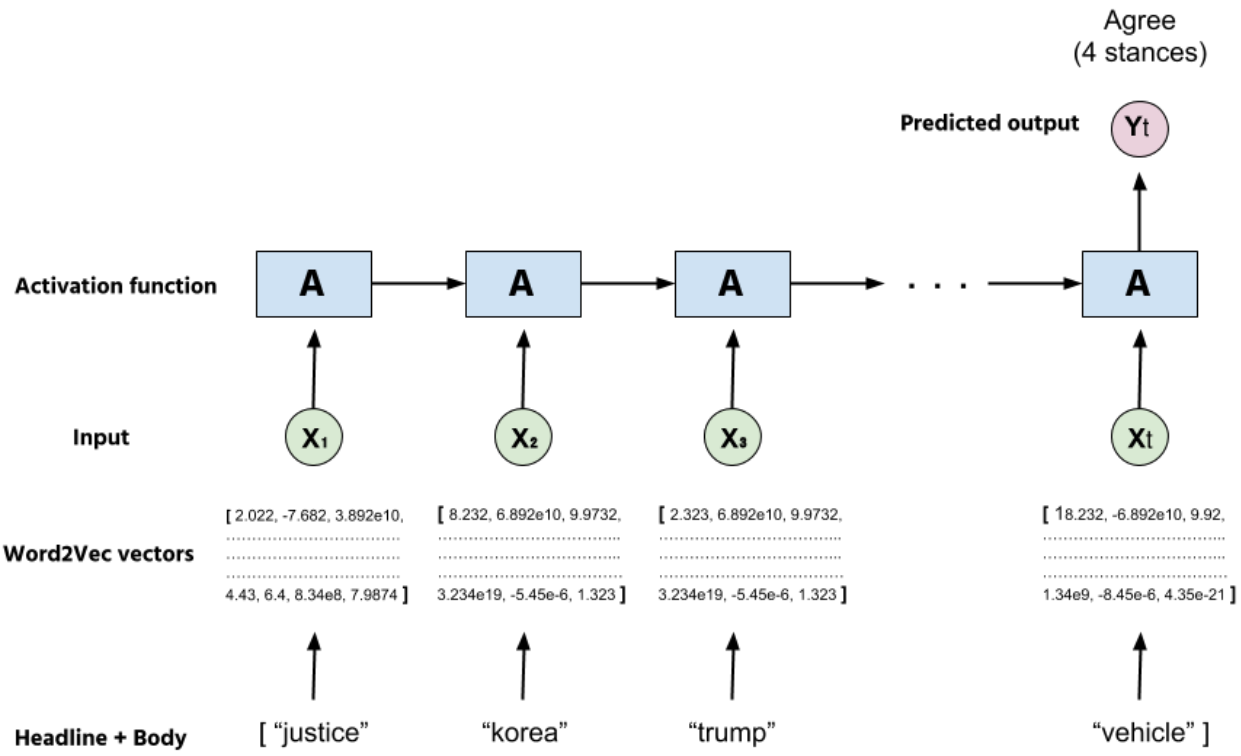

Fig. 3. Many to One RNN

3 Results & Discussion

In comparison to the 2-layer neural network TD-IDF implementation with BDR in the pilot study, the RNN Word2Vec model does provide an improved model performance where data sparisty is no longer an issue. And the current model is not overfitted due to Word2Vec and the structure of RNN. The 128 hidden size was chosen based on the conventions and the experiment tested by LO on the impact of number of hidden neurons to model performance and performance comparison of LSTM by Sato [7], [8]. The results from using *tanh* activation function without BDR, the 3-layer RNN shows a poor model performance as shown in figure 5. One important note is that the sequence length of testing corpus was different from the original trained corpus. As a result, the inputs including the sequence length of the testing dataset was padded by the differences of original minus the testing sequence of zeros on either side.

As shown in figure 6, the loss continues to decrease as it tries to find the local minima; Yet the model accuracy stopped improving after the first 50 epoch. The reason for the stablized model accuracy performance is because predicted output differs within the epoch. Thus, the model continues to predict the same class hence the stabled accuracy while the predicted output continues to compute the loss of the same class hence showing different losses.

Training the model without BDR estimated 120 minutes running on 2.9 GHz Intel Core i5 processor; while with BDR it estimated 45 minutes. Thus, BDR has certainly proven to not only increase the speed of the training but also reduce the number of unrelevant inputs. The results are shown in figure

In evaluating the goodness of the model with and without BDR, figure 6 hows the performance meaures of the trained model. As a result, RNN with BDR has proven higher recall, precision, accuracy, and F1. In detail, RNN with BDR achieved 7% higher precision ratio than without BDR; at least 3% more was predicted correctly in the RNN with BDR model. Although, the performance measures are lower than resarch done by Bajaj, where recall achieved was 0.56, precision was 0.91, and F1 was 0.70 [15]; RNN with BDR has proven a better performance model than 2-layer NN with BDR which resulted in an overfitted model.

RNN without BDR	Epoch	Loss	Accuracy
Vocabulary Size: 19628	1/500	62.0420	73.04%
	51/500	58.5125	73.04%
	101/500	56.4051	73.04%
	151/500	54.4891	73.04%
	201/500	52.7583	73.04%
	251/500	51.1922	73.04%
	301/500	49.7869	73.04%
	351/500	48.5393	73.04%
	401/500	47.4502	73.04%
	451/500	46.5219	73.04%

Fig. 4. RNN without BDR *tanh*

RNN with BDR	Epoch	Loss	Accuracy	Patterns Removed (Inputs)
Vocabulary Size: 19300	1/500	0.0136	6.80%	75
	51/500	0.0119	6.90%	867
	101/500	0.0111	44.73%	0
	151/500	0.0104	59.20%	340
	201/500	0.0098	66.85%	0
	251/500	0.0092	72.74%	0
	301/500	0.0087	76.85%	0
	351/500	0.0083	79.89%	0
	401/500	0.0078	82.22%	0
	451/500	0.0075	84.07%	0

Fig. 5. RNN with BDR *tanh*

	Recall	Precision	Accuracy	F1
RNN with BDR	0.2879057	0.259828	0.92769694	0.26029068
RNN without BDR	0.25	0.1826	0.86520004	0.21104947

Fig. 6. Performance Measures Training *tanh*

3.1 Evaluation Measures

A 4x4 multi-class confusion matrix is used as a evaluation measure where errors can be observed from the true class. The color frequency displays the instances that can range from 5 words up to 1000 words per a document. Thus, the maximum instances can go up to the vocabulary size multipled by the number of the maximum words in a document if the word exists in the vocabulary. Below is a confusion matrix of a 1200 inputs retrieved from the testing corpus with a total of 15405 vocabulary words of RNN with BDR.

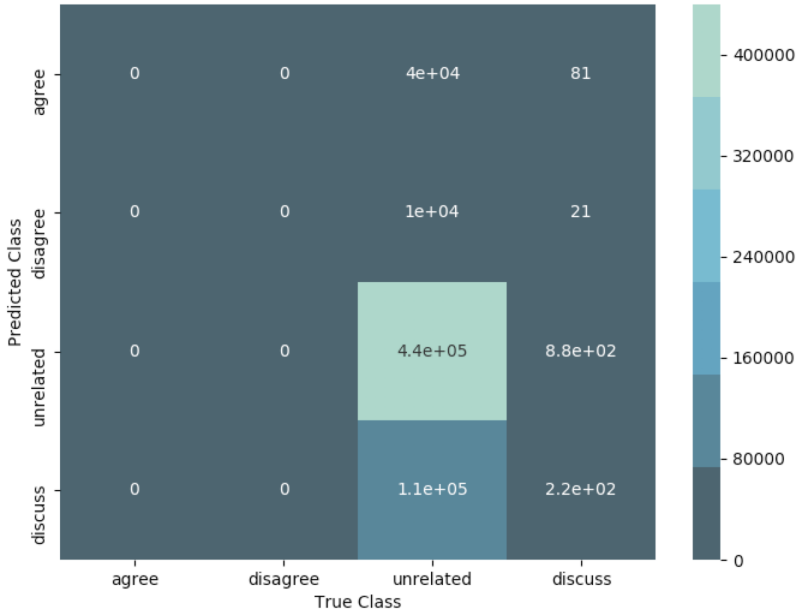

Fig. 7. Confusion Matrix Testing Corpus **1200 Inputs**

The diagonal shows the number of correct classification for each class. 0, 0, 4.4e+05, 1.1e+05 for the classes 'Agree', 'Disagree', 'Unreleated', 'Discuss', respectively. The model was able to successfully predict 'Unreleated' class with 4.4e+05 instances from the RNN model. In addition, at (1,4), true class was 'agree' but the model predicted 81 instances as class 'discuss'. Furthermore, at (3,4), the true class was 'unrelated' but the model predicted 8.8e+0.2 instances as 'discuss'. As a result of this experimental run, the model accuracy achieved at the end of 500 epoch was 73.29% as shown in figure 6.

	Epoch	Loss	Accuracy
	1/500	23.7469	18.08%
	51/500	11.8808	72.33%
	101/500	11.8808	72.87%
Total news articles **1200**	151/500	11.8808	73.05%
Total input neurons **2150**	201/500	11.8808	73.14%
Number of epoch **500**	251/500	11.8808	73.20%
	301/500	11.8808	73.23%
	351/500	11.8808	73.26%
	401/500	11.8808	73.28%
	451/500	11.8808	73.29%

Fig. 8. Model Performance Testing Corpus **1200 Inputs**

In figure 9, the performance measures of the model for the testing corpus is very low and one of the reasons is because given the low amount input data resulted in less outlier removal in comparison to a larger dataset. In addition, the results may be skewed given that the testing inputs had to be padded due to the unmatched sequence length for the original training corpus for sequence models like RNN.

	Recall	Precision	Accuracy	F1
RNN with BDR	0.045	0.2499	0.72	0.0723
RNN without BDR	0.0012	0.1432	0.5923	0.07627

Fig. 9. Performance MesuresTesting Corpus **1200 Inputs**

4 Conclusion

In conclusion, Word2Vec embeddings have proven to solve the overfitting and data sparsity issues with the previous research. Most importantly, sequence models such as RNNs do perform better with BDR in terms of resolving the bias-variance dilemma compared to a forwardfeed neural network with BDR. Given that RNNs is capability of receiving and outputting different input and output lenghts, RNN was the more suitable model architecture for the fake news stance detection problem.

4.1 Future Work

Word Embeddings
Word2Vec was used within the present research to represent words using dense vectors in a relatively low-dimensional space embeddings. However, research has shown that GloVe algorithm is more efficient in comparison with Word2Vec [17]. GloVe is a count-based model which is based on word occurrences in a textual corpus as opposed to Word2Vec which is a predictive model. GloVe model is built upon the construction of a co-occurrence 2D matrix from a training corpus where each matrix value is the frequency of the word co-occuring with another word [17]. In addition, GloVe also considers the factorization of the co-occurrence matrix in order to get vectors. The matrix factorization methods decompose large matrices that capture statistical information about a given corpus [18]. Meaning of words can be extracted from the co-occurrence probabilities; And given two words, the ratio of the co-occurrence probabilities with various probe words, the ratios help reduce noise by identifying relevant words from irrelevant words [18]. Yet, Word2Vec can only predicts words based on their context words which is dependable on the size of the window resulting in a less efficient results for the textual representation [17]. Previous research has shown that RNN with the use of GloVe have achieved higher precision than a feedforward neural network with GloVe on the FNC-1 datasets [15]. Unlike Word2Vec, GloVe produces a vector space by examining their various dimensions of difference instead of distance or angle between pairs of word vectors for similiarities [18]. As a result, it is important to further research on the effects of using GloVe model to represent texual datasets for future research.

Convolutional Neural Network Research has shown that Convolutional Neural Network (CNN) with max pooling and attention along with hyperparameters of an embedding size of 300, learning rate at 0.001, and hidden size at 100 can achiee a 0.97% in precision [15]. Moreover, the results of William Wang's research on the same task of fake news detection show that the CNNs outperformed all models including logistic regression classifier (LR), a support vector machine classifier (SVM), and a bi-directional long short-term memory networks model (Bi-LSTMs) with an accuracy of 0.270 on the heldout test set [23]. The research used a pretrained 300-dimensional Word2Vec embeddings from Google News. Thus, it would be worthy to further investigate with GloVe embeddings on CNNs to compare results and see the differences within the hyperparamters using BRR algorithm for noise removal for future work.

References

1. S. Tavernise, "As Fake News Spreads Lies, More Readers Shrug at the Truth", Nytimes.com, 2018. [Online]. Available: https://www.nytimes.com/2016/12/06/us/fake-news-partisan-republican-democrat.html. [Accessed: 20- May- 2018].
2. M. Barthel, A. Mitchell and J. Holcomb, "Many Americans Believe Fake News Is Sowing Confusion", Pew Research Center's Journalism Project, 2018. [Online]. Available: http://www.journalism.org/2016/12/15/many-americans-believe-fake-news-is-sowing-confusion/. [Accessed: 20- May- 2018].
3. "FakeNewsChallenge/fnc-1", GitHub, 2018. [Online]. Available: https://github.com/FakeNewsChallenge/fnc-1. [Accessed: 29- Apr- 2018].
4. S. Abeywardana, "Fake News Classifier (using LSTMs) Towards Data Science", Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/fake-news-classifier-e061b339ad6c. [Accessed: 26- May- 2018].
5. "Deep Learning with PyTorch PyTorch Tutorials 0.4.0 documentation", Pytorch.org, 2018. [Online]. Available: http://pytorch.org/tutorials/beginner/nlp/deep_learning_tutorial.html. [Accessed: 29- Apr- 2018].
6. J. Mody, "Sequence to Sequence with LSTM", Jackdermody.net, 2018. [Online]. Available: http://www.jackdermody.net/brightwire/article/Sequence_to_Sequence_with_LSTM. [Accessed: 26- May- 2018].
7. "How Does the Number of Hidden Neurons Affect a Neural Networks Performance", Chioka.in, 2018. [Online]. Available: http://www.chioka.in/how-does-the-number-of-hidden-neurons-affect-a-neural-networks-performance/. [Accessed: 27- May- 2018].
8. "Performance comparison of LSTM with and without cuDNN(v5) in Chainer", Chainer, 2018. [Online]. Available: https://chainer.org/general/2017/03/15/Performance-of-LSTM-Using-CuDNN-v5.html. [Accessed: 27- May- 2018].
9. R. Grosse, Cs.toronto.edu, 2018. [Online]. Available: http://www.cs.toronto.edu/ rgrosse/courses/csc321_2017/readings/L15
10. M. Barthel, A. Mitchelle, and J. Holcomb, "Many Americans Believe Fake News is Sowing Confusion", Pew Research Center, 2016.
11. Q. Zeng, Q. Zhou, "Neural Stance Detectors for Fake News Challenge."
12. R. Baly, M. Mohtarami, J. Glass, L. Marquez, A. Moschitti, and P. Nakov, "Integrating Stance Detection and Fact Checking in a Unified Corpus", Cornell University, 2018.
13. I. Augenstein, T. Rocktaschel, A. Vlachos, and K. Bontcheva, "Stance Detection with Bidirectional Conditional Encoding", In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016.
14. L. Peng, "Fake News Neural Networks: Bimodal Distribution Removal", ANU Biannual Bio-Inspired Computing Student Conference, 2018
15. S. Bajaj, "'The Pope Has a New Baby!' Fake News Detection Using Deep Learning", Stanford University, 2017.
16. A. Agren, C. Agren, "Combating Fake News with Stance Detection using Recurrent Neural Networks", University of Gothenburg, 2018.
17. M. Marwa, A. Chaibi, and H. Ghezala, "Comparative Study of word embeddings methods in topic segmentation", International Conference on Knowledge Based and Intelligent Information and Engineering Systems, pp. 340-349, 2017.
18. J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation", Conference: Conference: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 14. 1532-1543. 10.3115/v1/D14-1162.
19. J. Irvin, E. Chartock, and N. Hollander, "Recurrent Neural Networks with Attention for Genre Classification", Stanford University, 2016.
20. S. Pfohl, O. Triebe and F. Legros, "Stance Detection for the Fake News Challenge with Attention and Conditional Encoding."
21. Q. Zeng, Q. Zhou, "Neural Stance Detectors for Fake News Challenge."
22. Pennington, "GloVe: Global Vectors for Word Representation", Nlp.stanford.edu, 2018. [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed: 27- May- 2018].
23. Y. William ,"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection, pp. 422-426, 10.18653/v1/P17-2067, 2017.
24. Slade P., Gedeon T.D. Bimodal distribution removal. In: Mira J., Cabestany J., Prieto A. (eds) New Trends in Neural Computation. IWANN 1993. Lecture Notes in Computer Science, vol 686. Springer, Berlin, Heidelberg, 1993
25. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning", Nature, Macmillian Publishers, vol 521, pp. 436-444, 2015.
26. I. Goodfellow, Y. Bengio and A. Courville, Deep Learning. MIT Press, 2016.

Appendix

Confusion Matrix from 5000 inputs training corpus on RNN using *tanh* function without BDR

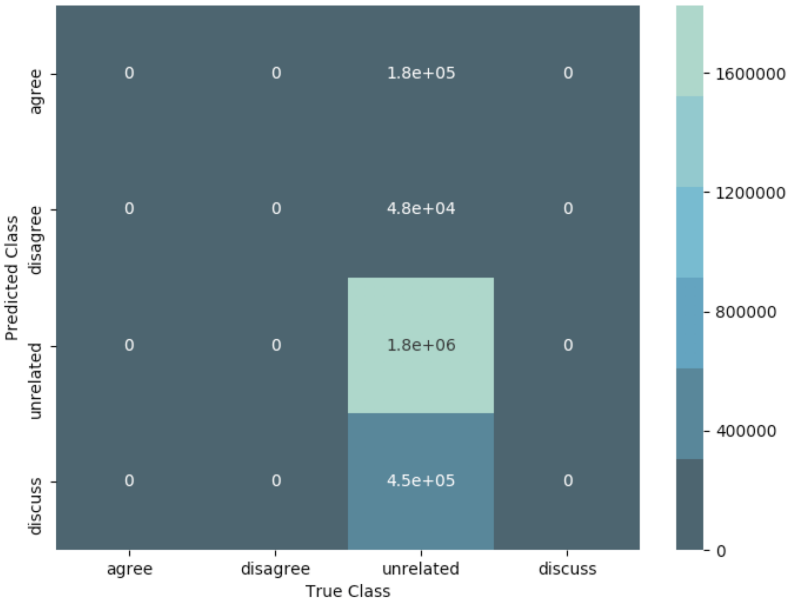

Fig. 10. Confusion Matrix **5000 Inputs**

Confusion Matrix from 5000 inputs training corpus on RNN using *tanh* function with BDR

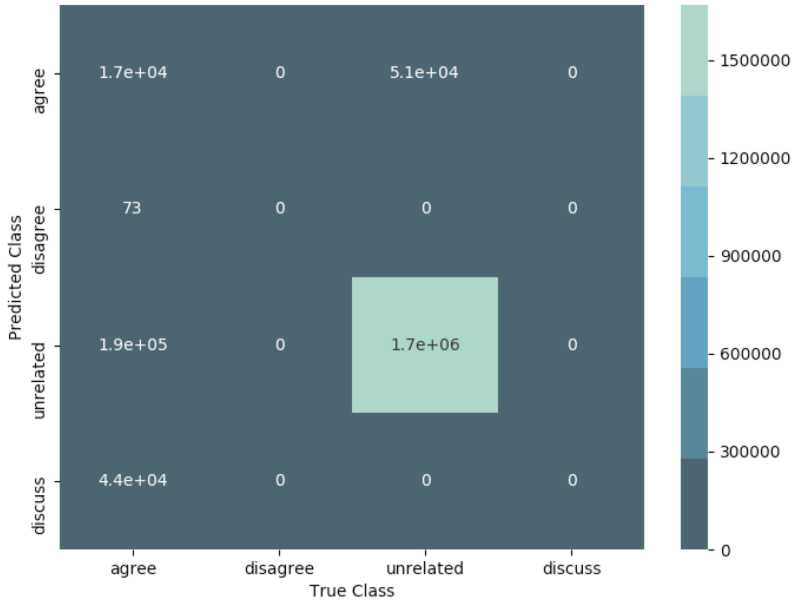

Fig. 11. Confusion Matrix **5000 Inputs**