Fake News Classification using neural networks and ensemble methods

Ashin George[*] PID A53247164
Arun Joseph[†] PID A53238815
Jose Joy[‡] PID A53230620
Unnikrishnan Sivaprasad[§] PID A53230407

ABSTRACT

Fake news is a major problem that has widespread social and political implications. In this paper, we analyze various existing techniques that are used for fake news detection. We propose a novel web-assisted ensemble model that smartly combines the output of a multilayer neural network and a real-time web search to accurately classify news. Our final model uses a combination of n-grams, TF-IDF , Non-negative Matrix Factorization and Latent Dirichlet Allocation as inputs to a Multilevel Perceptron. Based on the confidence level of the MLP we invoke a secondary classifier that makes use of real-time web search to classify the news based on a given headline. For this paper we also created a new labeled dataset for fake news classification, which we have named Beautiful Liar. This was combined with various other open source datasets to come up with a corpus of over 450K news articles and this final corpus was used to train and test our model. Our model was able to achieve an overall accuracy of around 86% with a BER of 0.13

Key Words: *Fake news, N-grams, TF-IDF, Latent Dirichlet Allocation, Non-negative Matrix Factorization, logistic regression, Random Forests, Neural Networks, Multi layer Perceptron, real time web assisted classification*

1 INTRODUCTION

News articles that are intentionally designed to mislead readers has grabbed public attention after the 2016 US presidential elections. The rising popularity of social media networks like Facebook, Twitter has made it easier to fabricate and propagate false news and vastly increased the reach and impact of such misinformations, converting them into tools of social control and divisive politics.

According to recent studies [1, 6, 8]:
a. 62 percent of US adults get news on social media
b. Fake news stories are more widely shared than true news stories
c. A large fraction of people who see fake news stories tend to believe them

Fake news generates economic gain and political power for some but this is done at great personal and social costs.

In this work, we explored the problems of fake news detection from a semantic point of view, trying to identify fake news by analyzing only the headline. We extracted features from the dataset using methods like n-gram models, TF-IDF, Non-negative matrix factorization(NMF) and Latent Dirichlet Allocation(LDA). We used the combination of these features with various classification models such as logistic regression, Support Vector Machines(SVM), random forest and Multilayer Perceptron(MLP) to classify fake news based on the extracted features.

This paper is organized as follows: Section 2 presents our data sources and the corpus of news articles. The Beautiful Liar dataset, which we created for this paper, is also described in that section. Section 3 describes the classification task that we are trying to solve in this paper. Section 4 examines the features we extract from the data and section 5 examines the various classification learning models. In section 6 we present the different approaches we have used for classification. We discuss relevant literature in section 7 and summarize our results in section 8.

2 DATASET

2.1 Data Aggregation

To obtain the data needed to train and test the model, we explored different sources. From Kaggle, we obtained the Getting real about fake news datasets, posted by Megan Risdal, which contains data scraped from 244 websites tagged as fake by the BS Detector Chrome Extension by Daniel Sieradski. From the UC Irvine machine learning repository, we obtained a news dataset called News aggregator dataset, which contains 423k news articles from various categories . We also accessed data from Politifact.com website, which is a website that a fact-checking website which rates the accuracy of correctness of the claims made by officials and others. The BBC news datasets, obtained from the Insight Project hosted by the University College Dublin consists of 3k news articles from BBC News [4]

The Liar Liar dataset is a collection of 12.8k statements from Politifact.com labelled on their truthfulness as pants-fire, false, barely-true, half-true, mostly-true, and true. We created an updated version of this dataset, which we are calling Beautiful Liar, that contains 15k articles. To create this updated dataset of news articles we scraped data from Politifact.com using Beautiful soup. These news articles were labelled as true, fake and ambiguous based on the politifact Truth-o-meter score. By combining the existing datasets along with our Beautiful Liar dataset we created a corpus of 454k articles

Data Sources

Source	type	size
Kaggle: Getting real about fake news	fake news	12999
UCI: News Aggregator Dataset	true news	422937
Politifact: Beautiful liar	mixed	14286
Insight project: BBC news database	true	2962
Combined		**453184**

2.2 Data Wrangling

To process the data into a uniform format, we removed punctuations, stop words, numbers and irrelevant characters from the headlines. The cleaned data were collated into a pandas data frame. Each of these entries were labeled with the ground truth by categorizing them into: true, fake and ambiguous', to make the corpus consistent with the data extracted from Politifact.com. The final corpus consisted of around 454,000 labelled news headlines with each headline containing an average of 10 words. To keep the problem tractable a dataset of 80,000 data points were collected from the corpus by stratified sampling to avoid sampling bias. This data was split into training, validation and test sets of sizes 60,000, 5000 and 15,000 respectively.

[*] e-mail: asg043@eng.ucsd.edu
[†] e-mail:arj035@eng.ucsd.edu
[‡] e-mail:jojoy@eng.usd.edu
[§] e-mail:usivapra@eng.ucsd.edu

Figure 1: Word Cloud for Fake True Ambiguous (left to right)

3 Classification Task

3.1 Task

The proliferation of fake news is a challenge facing the news industry today. Identifying whether a given article is a real news or not is a challenging task. This problem can be treated as identifying fake news articles as form of Needle-in-haystack problem. The task can also be treated as a classification problem of classifying the given news articles into actual news and fake news. In this paper, we are taking the second approach. We have modified the problem into a multi-class classification by adding a 'ambiguous news' class when the model identifies the given article to be similar to news articles labeled as ambiguous. News articles for which the model is not confident about its prediction are also labeled as ambiguous. The classification can be enhanced by using a second level classifier to predict these ambiguous news articles. We propose classifying these articles as true news or fake news based on their similarity with Google news results.

3.2 Evaluation Metrics

The optimal evaluation metric for the classification task depends on how the problem is modeled. Recall is the most useful metric for the needle-in-haystack formulation as the task is to identify all the fake news articles from the given set. When the task is formulated as a multi-class classification problem, as we have done in the paper, accuracy and Balanced Error Rate(BER) are more meaningful. Of these, BER is the most relevant metric. The number of fake news articles could be lesser than the number of true news articles. Focusing on accuracy would benefit a trivial classifier that predicts true news rather than a classifier that distinguishes between true news and fake news. In this paper, we report both the BER and accuracy for the different models.

3.3 Baseline Model

For this task, we chose a simple multinomial logistic regression as the baseline model, which is described in Section 6.1.
The different features used for the task are described in section 4 and the learning models in section 5 in the models described in section 6. The models were trained on the 60,000 entry training dataset and the parameters were tuned using the 5000 entry validation set. Each of these models were evaluated on the 15000 entry test set for BER and accuracy. We have observed that the performance on the validation set and test set closely match.

4 Features Used

4.1 N-Grams

N-grams is the continuous sequence of n words from a string of words. It represents a probabilistic model of language used to predict next likely word in a sequence. n-grams of size 1 are called unigrams and of size 2 are called bigrams. We extracted unigrams and bigrams from each class labels, with a constraint that only the n-grams which were present in no more than 90 % of the news articles were chosen.
We varied the number of n-grams that were used for classification and we determined the accuracy and BER in training, validation and test sets. Figure 2 summarizes this result. From figure 2 we can see that both accuracy and BER improves with increase in number of n-grams used, but the complexity of the model also increases with number of n-grams. Keeping running time in mind we have propose a method where we find the inflection point in accuracy and BER using the derivative plot. This point was seen to occur at around 600, and after this point increasing the number of n-grams, gives only marginal improvement in performance. Hence, the optimal number of n-grams was 600.

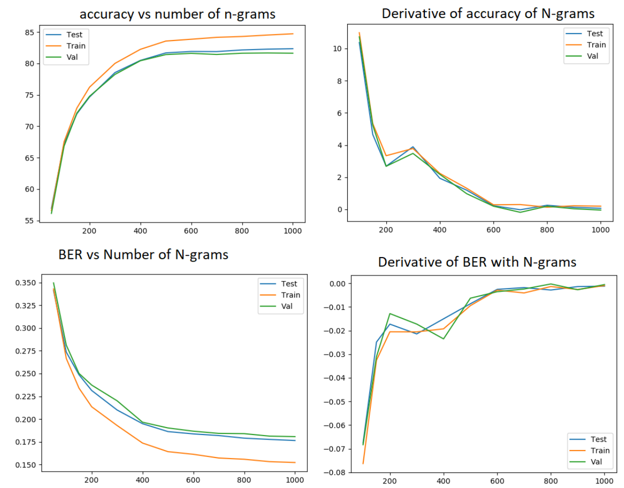

Figure 2: Variation of performance with number of N-grams

4.2 Term Frequency - Inverse Document Frequency (TF-IDF)

TF IDF or term frequency-inverse document frequency helps us to understand the importance of a word in a document. TF IDF value increases when the number of occurrences of the word in the document increases but it also takes into consideration the frequency of the word in different documents of a corpus. TF- IDF will automatically reduce the importance of certain words that are frequent in the news articles, which adds no value to a classifier. TF and IDF are calculated as per the equations,

$$tf(t,D) = count(t) \quad in \quad D$$

$$idf(t,D) = log\left(\frac{N}{|d \in D : t \in D|}\right)$$

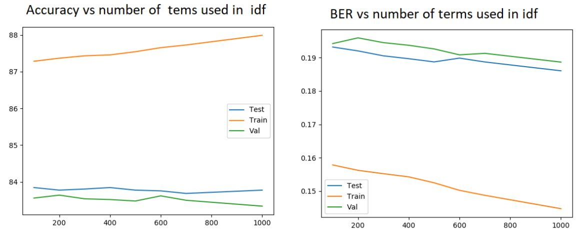

Figure 3: Variation of performance with number of TFIDF terms

4.3 Non negative matrix factorization(NMF)

Non-negative matrix factorization (NMF), is a group of algorithms used to factorize a matrix into two smaller matrices, with the property that all three matrices having nonnegative elements. The factorization is useful to find low dimensional representation of large matrices, and grouping similar entries together. So, in this paper, true news and fake news should be further apart in this low dimensional space, making it easier for the model to classify news. To factorize the matrix, we used frobenius norm.

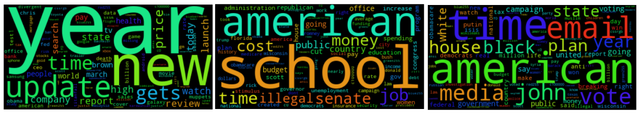

Figure 4: NMF Word Cloud True ambiguous false (left to right)

4.4 Latent Dirichlet allocation (LDA)

Latent Dirichlet allocation (LDA) is one of the generative statistical models in natural language processing, that allows sentences to be explained by unobserved groups, which identifies parts on the sentence belonging to a particular topic. This is very useful as news articles can be grouped into certain topics, which would possibly result in fake news becoming one topic and true news another, resulting in easier classification. These topics would be found in a process similar to unsupervised learning, which gave us an opportunity to find these unobserved topics among the news articles in our training dataset.

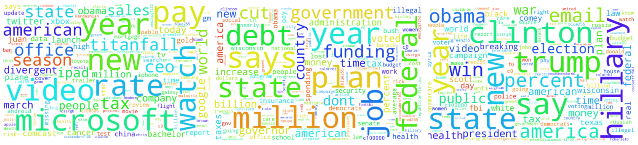

Figure 5: LDA Word Cloud True ambiguous false (top to bottom)

5 Classification Models

We used different classification models, namely Logistic regression, Support Vector Machines, Random Forest and Multi-Layer Perceptron.

5.1 Logistic regression

Logistic regression is a classification model which is a derivative of linear regression, that is used for binary classification tasks. The output from a linear regression model is transmitted to a logit function and using an appropriate threshold, can be used for binary classification. We used an ensemble of logistic regression models corresponding to each class (true, false and ambiguous news).

5.2 Support vector machines (SVM)

A Support Vector Machine (SVM) is a powerful machine learning model capable of performing linear or nonlinear classification, regression and outlier detection. SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier

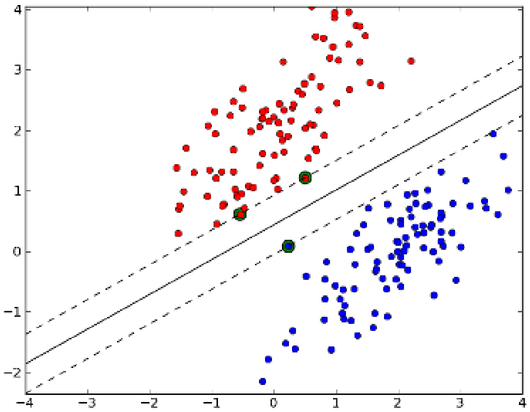

Figure 6: A visualization of SVM available in public domain.

In figure above the two classes can be easily separated with a straight line. If the training data is linearly separable as mentioned above, we can strictly impose that all instances above the line is in class A and everything below is in class B. This is called hard margin classification. But, if there is an item of class A below the line, in other words if the classes are not linearly separable we will be forced to use soft-margin classification. Soft-margin classification finds a good balance between keeping the distance between the dashed line maximum and limiting the margin violations or outliers. For this paper, we used a Linear SVM model, that uses a linear kernel, where the samples in space are separated by a hyper plane.

5.3 Random Forest

Random forest classifier is an ensemble model that uses a number of classification trees (weak classifiers) to create a strong model. Each tree predicts the class label of a feature sample provided, and the class label predicted by most number of trees is considered as the final prediction. The decision trees are generally trained using bagging methods and introduces extra randomness when growing trees. Each individual tree is trained using only a small subset of training samples, to reduce bias.
Since each tree encounters only subset of the training dataset, and there are multiple trees trained using multiple such subsets, trees are very diverse, which results in a lower variance, as the model make use of all such trees, generally yielding an overall better model. Another great quality of Random Forest is that they make it easy to measure the relative importance of each feature. We can measure feature importance by at the extent to which the tree nodes reduce impurity on an average. Random forest also does not overfit. Each tree can also be implemented using simple if else-if statements, making this model versatile for large datasets.

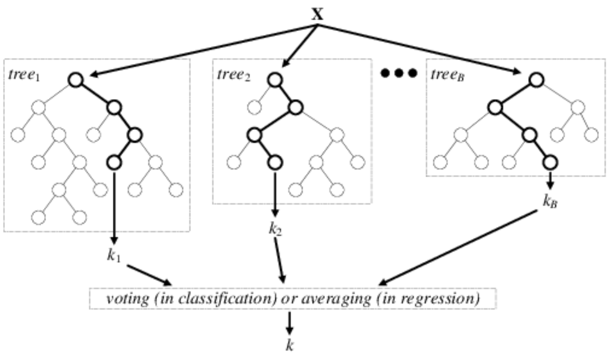

Figure 7: A visualization of Random Forest available in public domain.

5.4 Multilayer Perceptron

A multilayer perceptron (MLP) is an example of a feedforward artificial neural network. An MLP consists of at least three layers of nodes, where each layer except the output layer have weights and biases associated with it. Except for the input nodes, each node is a neuron utilizes a nonlinear activation function. MLP uses the revolutionary back propagation training method. For training, the algorithm feeds data to the network and computes the output of every neuron in each layer. Then it calculates the model's output error and it computes how much each neuron in the hidden layer contributed to each output neurons error. It then proceeds to compute how much of these error contributions came from each neuron in the previous hidden layer and so on until the algorithm reaches the input layer. Then the model slightly tweaks the connection weight to reduce the error. The ReLU function and hyperbolic tangent function are the most common non-linear activation function used in the model. Since multiple layers uses non-linear activation functions, its is very different from a linear perceptron.

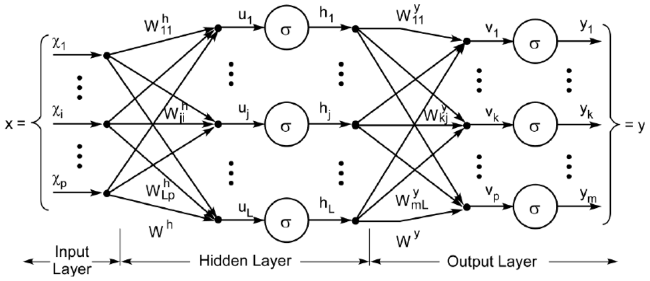

Figure 8: A visualization of Multilayer Perceptron.

6 APPROACHES

We used several features extracted using n-grams, tf-idf, non-negative matrix factorization and latent dirichlet allocation. We also used different classification models, namely Logistic regression, Support Vector Machines, Random Forest and Multi-Layer Perceptron [3, 7, 9].

6.1 Logistic Regression using n-grams

The baseline model was constructed using a simple multinomial logistic regression. The 250 most common unigrams and 250 most common bigrams in the entire training dataset were used as features in this model. The n-grams that were present in more than 90% of news articles were not used in this calculation as these are not likely to have any distinguishing information. This 500 feature model has been used to establish the hardness of the data set used.

6.2 Logistic Regression using n-gram classes

We approached the classification from the most basic sense first. We first separated our dataset into three classes, each representing true, false and ambiguous samples. We then found unique words (unigrams) and two letter combinations(bigrams) in each of these classes and stored them into dictionaries, along with the number of occurrences of these n-grams. We thus had three dictionaries for each class label, which had n-grams as the keys and number of occurrences as the values. From these, we selected few most frequent unigrams and bigrams from each class label, which were present in no more that 90 % of the news articles. We then tried to extract features from the news articles using these n-grams. The feature extraction method passed the news headlines through each n-gram dictionary corresponding to each class, and if a particular n-gram in the dictionary was present in the headlines, was given a value one at that particular column of feature matrix. We then used a logistic regression model to train three different logistic regression models to find if a news headlines is true, false and ambiguous respectively. For each model, the negative samples were same as the two other classes of news which was not considered (fake and ambiguous for true classifier, for example.). After training the model, we tested each news samples by passing it through the three separate models and the prediction was based on the relative confidence of predictions of the three models. The number of unigrams and bigrams were hyper parameters in the model, and so we created a validation set to find the optimum number of n-grams and it was found to be around 600 unigrams and 600 bigrams corresponding to each class label, which was a total of 3600 features. The model achieved an accuracy of around 81 % overall accuracy, and Balanced Error Ratio of 0.26. The confusion matrix corresponding to the model is given below. For the purposes of representation, we have omitted ambiguous class from the confusion matrix.

6.3 Linear SVM using n-grams and TFIDF

For this model, we used the tf-idf metric along with unigrams and bigrams. From the three datasets corresponding to the three classes, we extracted top 200 words that were found in no more than 90 % of news headlines. We first calculated the inverse document frequency (weightage of each word by considering its occurrence in a headlines in a dataset) of words from its corresponding dataset and then multiplied this idf with term frequency (number of times a word was present in the news headline). We had a total 600 features, corresponding tfidf values using words from the three classes. Each row of the tfidf matrix represents a particular news headline and each column represents presence of a unique word. For this model, we used an SVM classifier, which was trained on the above 600 features and the 3000 features corresponding to n grams mentioned in the previous case. The model had an accuracy of 80 % with a Balanced Error Ratio of 0.22. The confusion matrix of the model is given as follows.

Advantage : SVM penalises misclassifications and hence is very useful for classification tasks

Disadvantage : SVM doesnt scale well with large amount of training samples

6.4 Random Forest using NMF and n-gram

Non-negative matrix factorization is a technique that factorizes a large matrix into two smaller matrices, such that the product of the two matrices gives us the original matrix back. We used this technique to factorize the tf-idf matrix which represent the presence of certain words, which are weighted by the importance of that word in a dataset. Factorization yielded two matrices, the first one represented each news headline in with a low dimensional vector and the second one, which represents a low dimensional representation of latent topics. Now, the tf-idf vector corresponding to each news headline can be represented as product of low dimensional representation vector and another vector which represents a latent component among different topics. For this particular model, we chose the number of latent components as 25 which gave us a 25-dimensional row vector representation of a news headline in a 25 component space with dimensions equal to the number of words used to find tf idf which was 600. For training, we used NMF function in sklearn.decomposition module to create a model we then fitted it with the entire tf-idf matrix to generate the feature matrix. We then used this feature matrix and training labels which to train random forest model with 1000 classification trees where each tree depth was limited to 4. For testing, we used the NMF model to generate low dimensional model of the tf-idf model corresponding to testing samples. We obtained an overall accuracy of 78% with BER of 0.23.

Advantage : Random Forest has very low bias and variance and is faster to train than SVM

Disadvantage : If features used are redundant, the individual trees wont be trained properly and overall accuracy suffers

6.5 Random Forest using LDA and n-gram

For this model, we made use of latent dirichlet allocation which is a method that converts a string to a different space where similar samples correspond to one topic. We utilized the LDA function of sklearn module to extract features. We used a feature matrix that contained term frequency of certain words in the news headline to fit the LDA model. We then used the fitted model to generate low dimensional representations of new headline corresponding to training samples. The subspace has 30 dimensions. We used a matrix of such low dimensional representation to train a random forest model. We obtained an overall accuracy of around 79% with BER 0.21.

Advantage : LDA in theory is a much better model to find unobserved topics from news dataset

Disadvantage : The topics determined by LDA corresponding to true news might be very similar to that of false news

6.6 Multilayer perceptron using all features

For this model, we used features corresponding to n-grams, Non-negative matrix factorization and Latent dirichlet allocation to train a multilayer perceptron model. All the features extracted are given to the input layer of the multilayer perceptron classifier. The perceptron uses an artificial neural network having ten hidden layers with five nodes at each hidden layer. The final layer is a softmax classifier with three classes. To train the model we used the MLPClassifier function in the sklearn module, with a feature matrix that was the combination of all features used in the previous models. This model had an overall accuracy of 84 % with BER of 0.14

6.7 Google News Assisted Multi Level Perceptron

We propose a novel smart ensemble of similarity with Google News results and MLP results. Here we pass the headline through an MLP trained on the news articles from the training set, If the resulting prediction is either not confident or if news is predicted to be ambiguous, we use a second layer classifier, which finds the similarity with top results from Google news results for topics in the news headline.

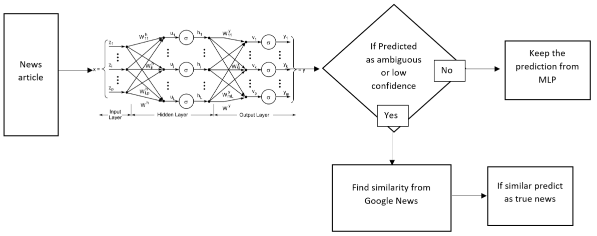

Figure 9: Flow chart of final model used

The inspiration for this model came from the following plot for MLP

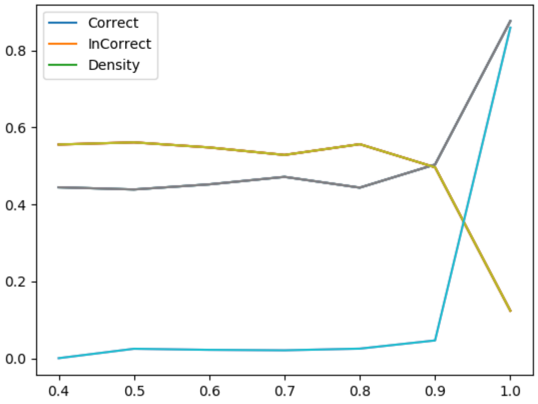

Figure 10: Variation of accuracy with confidence

Here we can see that the model is very accurate for predictions where confidence is higher than 0.9 and from the density plot, we can see that most of the news articles are predicted with very high confidence. The major chunk of errors that come from the predictions are due to the ones that have low confidence. So we use the second layer of classifier that measures the similarity of the given article with top 20 Google News articles related to the same topic, and if the article is found to be similar to these news results classify it as true, else was predicted as fake.

Using this method we got an accuracy of 86% and BER of 0.13

7 Literature

The dataset we used is mentioned in section 2. We used artificial neural networks and machine learning algorithms like SVM and random forest to classify news into fake news, true news or ambiguous news. Fake news detection is a very active problem. A fake news identifying competition called 'Fake News Challenge(FNC)' (http://www.fakenewschallenge.org) was organized by a consortium of academia and industry to detect fake news using machine learning, natural language processing and artificial intelligence features. The data derived from the Emergent Dataset created by Craig Silverman was used for this. FNC used stance detection as a method to detect fake news, by estimating the relative perspective (or stance) of two pieces of text relative to a topic, claim or issue.

TF-IDF, unigram and bigram are commonly used as features for fake news detection [3] [2]. These features is given as input to vari-

Table 1: Classification results for various approaches

	Basline: Naive Logistic	Logistic regression using n-grams	Linear SVM using n-grams and TFIDF	Random Forest using NMF and n-gram	Random Forest using LDA and n-gram	Multilayer perceptron using all features	Multilayer perceptron assisted by Google News
BER	0.41	0.26	0.22	0.23	0.21	0.14	0.13
Accuracy	0.70	0.81	0.80	0.78	0.79	0.84	0.86
TP	0.95	0.97	0.95	0.88	0.89	0.91	0.92
FP	0.74	0.48	0.39	0.34	0.33	0.21	0.19
FN	0.03	0.02	0.03	0.08	0.07	0.07	0.06
TN	0.23	0.49	0.56	0.57	0.59	0.72	0.71

ous classification models like SVM and Bayesian models to decide whether the news is true or fake. We have tried the same methods as well as neural network models like multi-layer perceptron which gave better result than SVM in our dataset. The winners of the FNC challenge used Convolutional Neural Networks on the headline and body text, represented at the word level using the Google News pre-trained vectors. The output of the CNNs was then sent to a MLP to classify the relation between the heading and the body. TF-IDF features and applied Singular-Value Decomposition to obtain a compact, dense vector representation of the headline and body. These features are then fed into Gradient Boosted Trees to predict the relation between the headline and the body. Jain, et al., [5] compare tweets with verified twitter News channels to label fake tweets. The Google News assisted MLP prediction follows a similar philosophy.

8 Results

We developed multiple approaches to classify news articles as true news, fake news or ambiguous news by analyzing the headline of the article. The multilayer perceptron neural network was the most successful model in correctly identifying and classifying fake news. The accuracy and BER for all these methods are given in Table 1. The proposed fake news detection model improved the classification accuracy by 22.9% compared to the baseline model, where as BER has become almost one third of the baseline, which represents significant improvement. The model we developed used a combination of n-grams, TF-IDF, Non-negative Matrix Factorization and Latent Dirichlet Allocation for feature extraction and used a multi-layer perceptron along with similarity with Google News results to classify the news articles. Our model has an accuracy of 86% and a BER of 0.13.

For future works, this model can be made to utilize both headlines and news body, to properly identify fake news and group news articles into multiple categories. This model can also be used to identify fake tweets and Facebook posts.

9 Conclusion

Fake news designed to spread misinformation and mislead people has gained interest in the public conscience following the 2016 US presidential elections and the discussion regarding how fake news were used to sway public opinion and impact the results are still on going. We created a model to classify news articles as fake, true or ambiguous based on the headline by using an ensemble method consisting of artificial neural network and various latent semantic analysis methods, which is assisted by real time news search from Google news.

Acknowledgments

The authors wish to thank Prof Julian McAuley. This work was done as part of CSE258: Recommender Sys & Web Mining.

References

[1] S. Bradshaw. Troops, trolls, and trouble makers: A global inventory of organized social media manipulation oxford computational propaganda research project: 21 ,2017, oxford university.

[2] C. Chen, K. Wu, V. Srinivasan, and X. Zhang. Battling the internet water army: Detection of hidden paid posters. *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, 2013.

[3] N. J. Conroy. Automatic deception detection: Methods for finding fake news. proceedings of the association for information science and technology, vol. 52, no. 1, 2015, pp. 14.

[4] D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering,proc. 23rd international conference on machine learning (icml'06), 2006.

[5] S. Jain, V. Sharma, and R. Kaushal. Towards automated real-time detection of misinformation on twitter. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.

[6] H. A. M.Gentzkow. Social media and fake news in the 2016 election. Journal of Economic Perspectives, May 2017.

[7] Y. Papanastasiou. Fake news propagation and detection: A sequential model. ssrn electronic journal, 2017, doi:10.2139/ssrn.3028354.

[8] C. Silverman and J. Singer-Vine. Most americans who see fake news believe it, new survey says. BuzzFeed, December 2016.

[9] Z. Zhang. Short text classification using latent dirichlet allocation, journal of computer applications, vol. 33, no. 6, 2013, pp. 15871590.