Fake News Detection: Using An Ensemble Approach

Viveksinh Solanki
vsolank3@stevens.edu
Stevens Institute Of Technology
Hoboken, New Jersey

1 INTRODUCTION

Fake news detection is the ongoing research area under the Natural Language Processing (NLP) domain. In simpler terms, fake news can be defined as any information which is misleading and verifiably false. Fake news can be harmful as it can propagate conspiracy or mistrust.

In this paper, we are borrowing methods from Machine Learning literature and applying those to automate the classification of fake news.

2 BACKGROUND

Fake news can be defined as any news with false information and which is created with the intent of misleading people. But, still, there is no widely accepted definition of fake news yet.

Fake news can be harmful in a variety of ways. First, fake news can break the authenticity balance of the news ecosystem. Second, fake news changes the way people interpret and respond to real news. For example, in the past couple of years, India has faced many mob lynching incidents due to fake news spreading on one of the top social media platforms, WhatsApp. To help mitigate the negative effects caused by fake news, it's critical that we develop methods to automatically detect fake news.

Today, most of the news is spread and consumed from social media than traditional news agencies, as it is more convenient for users. Because of this, social media is majorly responsible for disseminating fake news to the mass population.

Traditional approaches to fighting fake news include human annotators who manually check all the news and decide its genuineness based on the available truthful resources. Hence, the traditional approach of fake news detection is way more time consuming and tedious. By borrowing the methods from Machine Learning (ML) and NLP, we can automate the task of fake news detection, which will be way faster and cheaper with a little trade-off for accuracy.

3 RELATED WORK

For automatic fake news detection problem, there are mainly two types of approaches, which have been followed widely: linguistic-based approaches and fact-checking based approaches. In the later approaches, fake news is compared with the news from multiple reliable resources for classification[1]. We will be following the linguistic-based approach, which includes utilizing text properties as features for machine learning (ML) models.

The linguistic approach has yielded promising results in differentiating satire from real news[7]. Kai Shu[8] has proposed tri-relationship, which is the relationship among publishers, news pieces, and users, as baseline features for machine learning models to classify fake news. Bennett Kleinberg and team[6] have used linear Support Vector Machines(SVM) classifier with ngrams, punctuations, readability, syntax and Psycholinguistic features as different feature sets. This approach[11] uses text and image-based convolutional neural networks(TI-CNN) with news title, text, and image as features for fake news detection. Also, there has been a comprehensive study[4] done in which authors have surveyed different Natural Language Processing(NLP) based approaches on multiple publicly available datasets with different neural network and non-neural network models.

Table 1: The LIAR dataset statistics

Dataset statistics	
Training set size	10,269
Validation set size	1,284
Testing set size	1,283
Average statement length	17.9

Table 2: Umich dataset statistics

Dataset	Label	Entries	Total Words
FakeNewsAMT	Fake	240	31,990
	Legitimate	240	33,378
CelebritySet	Fake	250	39,440
	Legitimate	250	70,975

4 DATASETS

LIAR[1]: This dataset[10] is collected from a Pulitzer Prize-winning fact-checking website POLITIFACT.COM[2] through its API. It includes 12,836 human-labeled short statements, which are sampled from various contexts, such as news releases, TV or radio interviews, campaign speeches, etc. Each statement is evaluated by a POLITIFACT.COM editor for its truthfulness. The labels for news truthfulness are fine-grained multiple classes: pants-fire, false, barely-true, half-true, mostly true, and true. Dataset statistics are given in table 1. Also, LIAR includes many metadata columns like speakers, their party affiliation, current job, home state and credit history.

FakeNewsDatasets[3]: There are two datasets obtained from the University of Michigan[6]: FakeNewsAMT and CelebritySet.

- FakeNewsAMT: This dataset is created by first collecting legitimate news belonging to six different domains (sports, business, entertainment, politics, technology, and education).

[1] https://www.cs.ucsb.edu/~william/data/liar_dataset.zip
[2] http://www.politifact.com/
[3] http://web.eecs.umich.edu/~mihalcea/downloads/fakeNewsDatasets.zip

The legitimate news was obtained from a variety of mainstream news websites predominantly in the US such as the ABCNews, CNN, USAToday, NewYorkTimes, FoxNews, Bloomberg, and CNET among others. Fake versions of the legitimate news items were generated by crowdsourcing via Amazon Mechanical Turk (AMT). There is a total of 240 legitimate and 240 fake news.

- CelebritySet: The data was collected from online magazines such as Entertainment Weekly, People Magazine, RadarOnline, among other tabloid and entertainment-oriented publications. The data were collected in pairs, with one article being legitimate and the other fake. To determine if an article is fake or legitimate, gossip-checking sites such as "GossipCop.com" was used, and it was cross-referenced with information from other entertainment news sources on the web. There is a total of 250 legitimate and 250 fake articles.

5 APPROACH

Fake news detection is considered a classification task that predicts news as fake or not fake. We present our solution to apply 3 different machine learning models to all 3 datasets and get results. Previously, James Thorne[9] and the team have utilized a stacked ensemble approach for stance classification task and they got some good results. Hence, we have thought to get the final classification based on the voted classifier of all 3 models. The architecture diagram for the ensemble is given in figure 1.

Finally, we will compare the results from both types of experiments to identify the best model. In our case, we are dealing with text data so our features will be bag-of-words (TF-IDF) and word embeddings. For classification tasks, the most widely used metrics in machine learning are Precision, Recall, F1-score, and Accuracy. We are going to use these metrics for evaluating our models. These metrics will help in the understanding overall performance of our algorithms.

6 EXPERIMENTAL DESIGN

After data cleaning and preprocessing, we had 5721 real samples and 4455 fake samples remaining in LIAR dataset. The number of samples for FakeNewsAMT and Celebrity set were the same as given in the dataset description.

6.1 Individual Models

We decided to use 2 machine learning models: 1) Multinomial Naive Bayes (MNB) 2) Support Vector Machine (SVM) and 1 deep learning model: Convolutional Neural Network (CNN) for our classification task. We chose the support vector machine because it is considered one of the best models for classification tasks as it discriminates based on the largest margin boundaries. From the family of deep neural networks, we chose CNN as it has performed quite good for some of the natural language processing tasks. To compare the performance of SVM and CNN to some baseline we chose Multinomial Naive Bayes as our baseline model.

6.1.1 SVM & MNB. : For all 3 datasets first, we performed the grid search to find the best parameters for TF-IDF vectors. We created the TF-IDF matrix using the best parameters obtained from a grid search. For the LIAR dataset, we feed these features directly into SVM and MNB models to get the classification. For FakeNewsAMT and CelebritySet, we used 5-fold cross-validation to get the classification.

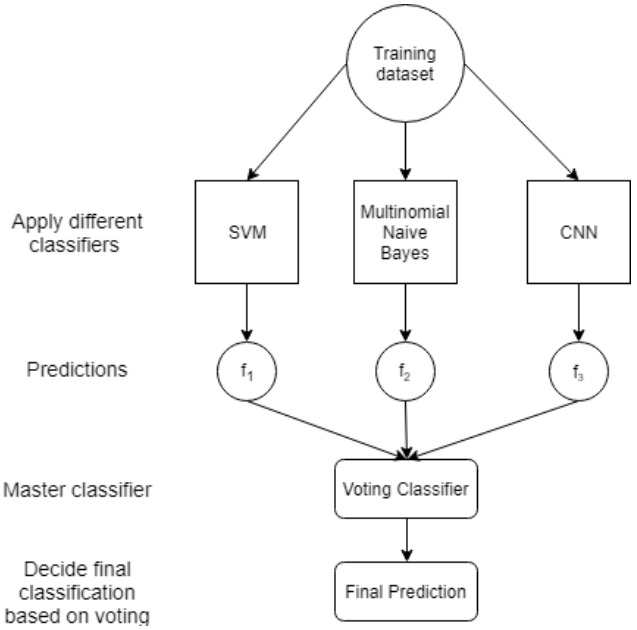

Figure 1: An ensemble approach using three classifiers: Multinomial Naive Bayes, Support Vector Machine, and Convolutional Neural Network

6.1.2 CNN. : To train the CNN model, first, we created word embeddings using a pre-trained GloVe[4] [5] model with 100-dimensional vectors. For layers and parameter settings refer tables 3 and 4. We tuned hyperparameters based on multiple experiments with a number of combinations of hyperparameters. We used rectified linear unit (ReLU) as the activation function for improved performance. We utilized the dropout layer to avoid overfitting. Along with the convolution and sampling (MaxPooling) layer, we used the output of GlobalMaxPooling layer as the input of a dense layer. Because GlobalMaxPooling is commonly used for natural language processing related classification tasks. To further avoid overfitting, we used early stopping on the validation set with parameter patience = 7. Keeping the value of patience high allows the model to not stop too early.

6.2 Ensemble of all 3 classifiers

The architecture for the ensemble is given in figure 1. After getting classification results for individual models, we performed majority voting to get the final voted classification. We kept the same parameter settings as in individual model experiments for all 3 models.

[4]https://nlp.stanford.edu/projects/glove/

Table 3: CNN Layers

Layers
Input
Embedding
Dropout
3 Convolution layers with 'relu' activation
2 MaxPooling layers
1 GlobalMaxPlooing
1 Dense layer with 'relu' activation
Dropout
1 Dense layer with 'sigmoid' activation

Table 4: CNN Configurations

Configs.
Loss = Binary Cross-Entropy
Optimization algorithm = Adaptive Learning Rate
Metric = Accuracy

6.3 Other experiments

We extracted word count, punctuation count, and sentence count before doing any kind of cleaning or preprocessing. We believed these features would allow us to improve the classification performance of the models. But after performing various experiments, we got negative outcomes. Every model gave lower performance while using these features. We also tried Doc2Vec[2] from Gensim[5] python package. Doc2Vec implements paragraph vector, an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents. Authors have mentioned that paragraph vectors outperform bag-of-words models as well as other techniques for text representations. But when we tried it on our datasets, paragraph vectors didn't improve performance for any of the models.

7 EXPERIMENTAL RESULTS AND ANALYSIS

For our problem statement, we care more about fake(positive) labels than real(negative) labels. Hence, accuracy will never be a good metric to measure model performance. we have compared model performance mainly based on their capability of correctly identifying more number of fake labels. Hence, we considered the true positive rate(recall) of positive class for comparison.

7.1 LIAR

We can see classification reports for CNN, SVM, and MNB in table 7, 8 and 9 respectively. As we can notice CNN has a higher true positive rate (59%) for the positive label(fake) as compares to SVM and MNB. The LIAR dataset is very big as compared to the other two datasets. We believe this to be the reason for CNN to perform better than the other two models.

Table 7: LIAR : CNN classification report:

[5]https://radimrehurek.com/gensim/models/doc2vec.html

Table 5: CNN accuracy scores when using word embeddings as features

Dataset	Accuracy(%)
LIAR	55.24
FakeNewsAMT	51.66
CelebritySet	57.6

Table 6: SVM and MNB results with TF-IDF as features (all scores are in %)

Model	Dataset	Accu	f1	precision	recall
SVM	FakeNewsAMT	63.33	63.66	63.76	64.58
	CelebritySet	78.8	78.03	82.13	74.8
MNB	FakeNewsAMT	61.87	63.25	61.27	66.25
	CelebritySet	79.80	78.71	84.41	74.4

	precision	recall	f1-score	support
0	0.62	0.53	0.57	711
1	0.49	0.59	0.53	547
micro avg	0.55	0.55	0.55	1258
macro avg	0.56	0.56	0.55	1258
weighted avg	0.56	0.55	0.55	1258

LIAR : SVM: Accuracy = 60.09%

Table 8: LIAR : SVM classification report:

	precision	recall	f1-score	support
0	0.64	0.67	0.66	711
1	0.54	0.51	0.52	547
micro avg	0.60	0.60	0.60	1258
macro avg	0.59	0.59	0.59	1258
weighted avg	0.60	0.60	0.60	1258

LIAR : MNB : Accuracy = 62.48%

Table 9: LIAR : MNB classification report:

	precision	recall	f1-score	support
0	0.65	0.74	0.69	711
1	0.59	0.47	0.52	547
micro avg	0.62	0.62	0.62	1258
macro avg	0.62	0.61	0.61	1258
weighted avg	0.62	0.62	0.62	1258

7.2 FakeNewsAMT

Here, we are comparing models based on their overall recall (macro average of recall in case of CNN). Results for the FakeNewsAMT set are displayed in tables 6 and 10. MNB model gave the highest recall among all 3 models for the FakeNewsAMT set.

Table 10: FakeNewsAMT : CNN classification report:

	precision	recall	f1-score	support
0	0.65	0.35	0.45	69
1	0.46	0.75	0.57	51
micro avg	0.52	0.52	0.52	120
macro avg	0.55	0.55	0.51	120
weighted avg	0.57	0.52	0.50	120

7.3 CelebritySet

Same as the FakeNewsAMT set, we have considered overall recall as our performance comparison metric. Results for CelebritySet are given in table 6 and 11. As we can notice from table 6, SVM and MNB models almost gave the same recall. But CNN has a very low macro average of recall of 57%. It has been shown that neural networks require a huge amount of data to learn underlying patterns. In the case of CelebritySet and FakeNewsAMT set, we have very fewer examples to feed into CNN. Hence, CNN couldn't beat traditional models.

Table 11: CelebritySet : CNN classification report:

	precision	recall	f1-score	support
0	0.63	0.61	0.62	71
1	0.51	0.54	0.52	54
micro avg	0.58	0.58	0.58	125
macro avg	0.57	0.57	0.57	125
weighted avg	0.58	0.58	0.58	125

7.4 Ensemble of classifiers

Results for an ensemble of classifiers are given in tables 12, 13 and 14. For LIAR and FakeNewsAMT datasets, the voted classifier of all 3 models gave lower performance than individual models. For the CelebritySet, the voted classifier performed almost the same as SVM and MNB, which implies CNN didn't add much value in the ensemble. Hence, for our chosen datasets, the ensemble approach is not that much good.

LIAR : Ensemble of classifiers :
Accuracy = 61.60%
Table 12: LIAR : classification report:

	precision	recall	f1-score	support
0	0.65	0.71	0.68	711
1	0.57	0.50	0.53	547
micro avg	0.62	0.62	0.62	1258
macro avg	0.61	0.60	0.60	1258
weighted avg	0.61	0.62	0.61	1258

FakeNewsAMT : Ensemble of classifiers :
Accuracy = 45%
Table 13: FakeNewsAMT : classification report:

	precision	recall	f1-score	support
0	0.62	0.12	0.20	69
1	0.43	0.90	0.58	51
micro avg	0.45	0.45	0.45	120
macro avg	0.52	0.51	0.39	120
weighted avg	0.54	0.45	0.36	120

CelebritySet : Ensemble of classifiers :
Accuracy = 72%
Table 14: CelebritySet : classification report:

	precision	recall	f1-score	support
0	0.85	0.62	0.72	71
1	0.63	0.85	0.72	54
micro avg	0.72	0.72	0.72	125
macro avg	0.74	0.74	0.72	125
weighted avg	0.75	0.72	0.72	125

The performance of CNN for each dataset is displayed in figures 2, 3, and 4. We have plotted two plots for each dataset: 1) model accuracy vs the number of iterations 2) model loss vs the number of iterations.

8 CONCLUSION

As we saw from the results ensemble of classifiers didn't work for the chosen datasets. Individual models performed better than the ensemble. To improve the performance of CNN, we need to have

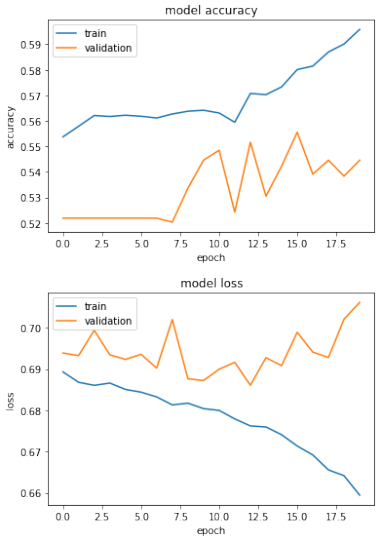

Figure 2: LIAR: CNN plots

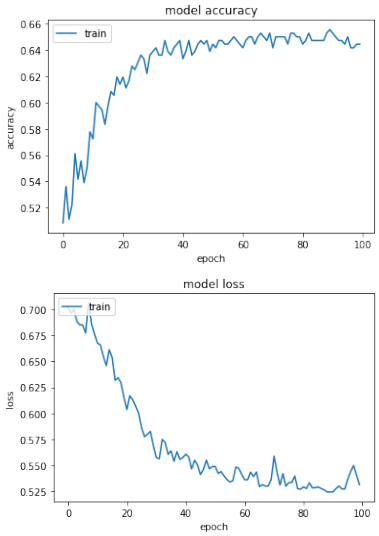

Figure 3: FakeNewsAMT: CNN plots
Due to the smaller size of dataset, we didn't separate validation set

larger datasets, otherwise, CNN won't be able to learn underlying patterns from the corpus.

The most important finding was that fake news detection solely based on machine learning and natural language processing methods is not robust at all. We performed many experiments with the same hyper-parameter combinations for all 3 datasets. But we couldn't get the same performance for any model. We also performed a quick experiment of applying given models on other datasets, which are not mentioned in this report, just to understand model performance for different datasets. We found out that model performance totally depends on the dataset. Hence, if we train the model on one fake news dataset and then use this trained model

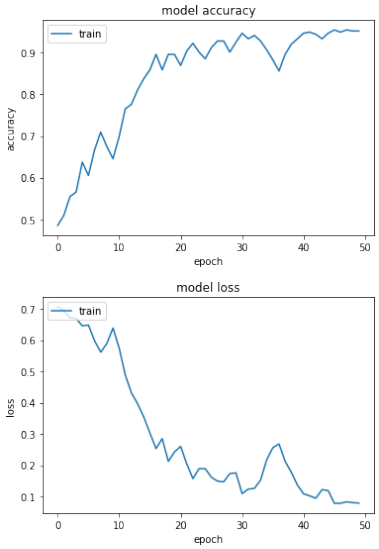

Figure 4: CelebritySet: CNN plots
Due to the smaller size of dataset, we didn't separate validation set

to test on totally different fake news dataset then it will perform poorly. So, we conclude that fake news detection systems should not solely be based on machine learning methods.

9 FUTURE WORK

We used GloVe to extract word embeddings. Basically, the GloVe model is trained on Wikipedia. On the other hand, Word2Vec[3] is trained on a very large corpus of news. As we are dealing with a corpus of news only, we believe that word embeddings created using the Word2Vec model might assign better context for words and eventually it might help CNN better. We have noticed that most of the fake news articles use exaggerated tone/emotion. So, it would be interesting to extract tone/emotion from the given text and utilize it as a new feature. It might perform better than just using sentiments.

Based on our findings, we propose a new approach for robust classification of fake news and real news. There should be some sort of automated evidence checking mechanism in the fake news detection pipeline. By evidence checking, we mean that facts from the given text should automatically be validated with similar text from reliable sources.

REFERENCES

[1] Niall J. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology* (2015), 1–4.
[2] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* abs/1405.4053 (2014). arXiv:1405.4053 http://arxiv.org/abs/1405.4053
[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[4] Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A Survey on Natural Language Processing for Fake News Detection. *CoRR* abs/1811.00770 (2018).
[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural*

Language Processing (EMNLP). 1532–1543. http://www.aclweb.org/anthology/D14-1162

[6] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. *COLING* (2018).

[7] Victoria L. Rubin, Niall J. Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. *Proceedings of NAACL-HLT* (2016), 7–17.

[8] Kai Shu, Suhang Wang, and Huan Liu. 2017. Exploiting Tri-Relationship for Fake News Detection. *arXiv preprint arXiv:1712.07709* (2017).

[9] James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. 2017. Fake news stance detection using stacked ensemble of classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. Association for Computational Linguistics, Copenhagen, Denmark, 80–83. https://doi.org/10.18653/v1/W17-4214

[10] 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

[11] Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S. Yu. 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. *CoRR* abs/1806.00749 (2018).