Texas A&M University-San Antonio

Digital Commons @ Texas A&M University-San Antonio

Computer Information Systems Faculty Publications

College of Business

8-2022

Fake News Detection on Social Media: A Word Embedding-Based Approach

Muammer Eren Sahin

Chunyang Tang

Mohammad A. Al-Ramahi

Follow this and additional works at: https://digitalcommons.tamusa.edu/cis_faculty

Part of the Computer Sciences Commons

Fake News Detection on Social Media: A Word Embedding-Based Approach

Completed Research

Muammer Eren Sahin
Texas A&M-San Antonio
msahi01@jaguar.tamu.edu

Chunyang Tang
Texas A&M-San Antonio
ctang01@jaguar.tamu.edu

Mohammad Al-Ramahi
Texas A&M-San Antonio
Mohammad.Abdel@tamusa.edu

Abstract

The rapid development of social media, together with the large number of user-generated content on them, has not only connected an unprecedented number of people together to do good stuff, but also has provided convenient platforms to spread misleading pieces of information such as fake news. Existing research has attempted to leverage machine learning to automatically classify fake news. In this paper, we extend such literature by proposing an approach that utilize word embedding and Long Short-Term Memory (LSTM) neural network algorithm. Unlike existing studies, we used two publicly available datasets of news articles to evaluate the proposed model. The results demonstrated the effectiveness of our model against the baseline machine learning models with accuracy of 99% and 96% using the first and second datasets respectively. These comparatively better results and effectiveness compared to existing models demonstrate that pre-trained word embedding models play a significant role in the fake news detection.

Keywords

Fake news, neural network, LSTM, word embeddings

Introduction

The rapid development of social media, together with the large number of user-generated content on them, has not only connected an unprecedented number of people together to do good stuff, but also has set up convenient platforms for malicious individuals, businesses, and political parties to spread incorrect, inaccurate, or misleading pieces of information, i.e., fake news (Ghosh and Shah 2018). Similar research topics (fake Amazon review, for example) have been proposed and analyzed long before the 2016 U.S. presidential election campaign, when the fake news problem played a major part. A growing number of research articles are published every year after 2016.

The fake news spread uncontrollably on social media platforms as fast as the speed of light and has caused significant economic, political, and social damage to the victims. One could argue that, in some ways a decentralized supply of information is helping to eliminate media biases, however, everyone has also come to the same conclusion that a growing number of fake news has to be stopped (Ahmed 2017).

Given the fact that fake news generation rate is inherently faster than the rate at which human fact checkers can verify or debunk them (Ahmed et al. 2021), it is impossible to only rely on human effort to fight fake news. Computers come in handy this way. If we can build a highly reliable, trusted, and highly efficient computer program that automatically classifies fake news from their real counterparts, it would cost-effectively contribute greatly to the greater good of societies. Therefore, a variety of proposals and research experiments have been accomplished in this area by computer scientists, with machine learning algorithms being the most popular (Al-Ramahi and Alsmadi 2021; Al-Ramahi and Alsmadi 2020; Choudhary and Arora 2021; Islam et al. ; Meneses Silva et al. 2021; Sharma et al. 2021).

Fake news could be defined as pieces of information that are inaccurate, incorrect, or misleading and are generated with explicit or implicit intent to manipulate the readers' view on certain targets. The fake news classification problem is, therefore, a problem that asks one simple question: given certain text, can we determine, with enough accuracy and within reasonable time and resource spent, that the text is fake or not. Practically we can benchmark a candidate machine learning algorithm solution by running it against a huge number of known real news and/or fake news, then calculating the accuracy, precision or F1.

Prior research (e.g., Granik and Mesyura 2017; Jadhav and Thepade 2019; Kurasinski 2020; Singh et al. 2021) investigated different machine learning algorithms to classify fake news. In this research, we aim to complement such research by proposing an automatic approach for fake article news detection. Given the promising results of using deep learning techniques in text classification, we developed our model using word embedding and Long Short-Term Memory (LSTM), a type of recurrent neural network (RNN), that is capable of learning dependencies in textual sequences. Hence, capturing the semantic relationship between words. The performance of our proposed model has been demonstrated using two different large scale news articles datasets, where the accuracy of 99% and 96% has been achieved. Our proposed approach shows comparatively better results comparing with existing models, which makes word embeddings a promising approach for accurate detection of fake news.

Related Work

The problem of fake news is increasing day by day due to the ease of access to the internet. Researchers have been using a variety of tools to fight against fake news issues for over a long time (Shu et al. 2017). Traditional ways of telling fake news are by using weapons from the natural language processing arsenal, including bag of words, n-grams and semantic analysis. Newer approaches in this field have seen the application of various machine learning algorithms including Decision Tree, Support Vector Machine, Naive Bayes, Logistic Regression and Neural Networks (Ahmed 2017).

Combining term frequency and support vector machine algorithms, a researching team (Ott et al. 2011) was able to get 86% accuracy when dealing with fake reviews of hotels against real reviews. Part of Speech (POS) as well as Linguistic Inquiry Word Count (LIWC) are also popular among natural language processing researchers. When building models with Sparse Additive Generative Model (SAGE), LIWC and POS are utilized with final accuracy about 64% (Li et al. 2014).

K nearest neighbor (KNN) also can play a part in the game against fake news. There have been attempts to use the algorithm (Kesarwani et al. 2020) to classify fake news on social media.

Recently, the promising results of deep learning techniques in text classifications has attracted researchers' attention to apply deep learning in fake news detection (e.g., Balwant 2019; Kaliyar et al. 2020; O'Brien et al. 2018; Vyas et al. 2021; Yang et al. 2018). A paper in 2020 (Kaliyar et al. 2020) employed deep learning neural networks to do the job with good performance over 90%.

One major obstacle remains. The source of good data with true news and fake news already classified by humans is still lacking. This requires a huge amount of work (Ahmed, 2017). As fake news generation mechanisms rapidly change, it is imperative that a good quality 'gold standard' dataset can be provided to global researchers.

Experiment

Datasets

First Dataset

The first news article dataset[1] used includes two comma separated value files, a *True.csv* for real news and a *Fake.csv* for fake news. The *True.csv* and *Fake.csv* each consist of 4 different columns with approximately 20,000 instances (Table 1). As one of the first steps to understand the data, we run some code checking the null values and the statistics of the features. Below are the word clouds from these two files.

	Size	Feature names
True.csv	21417 rows x 4 columns	['title', 'text', 'subject', 'date']
Fake.csv	23481 rows x 4 columns	

Table 1. First Dataset Description

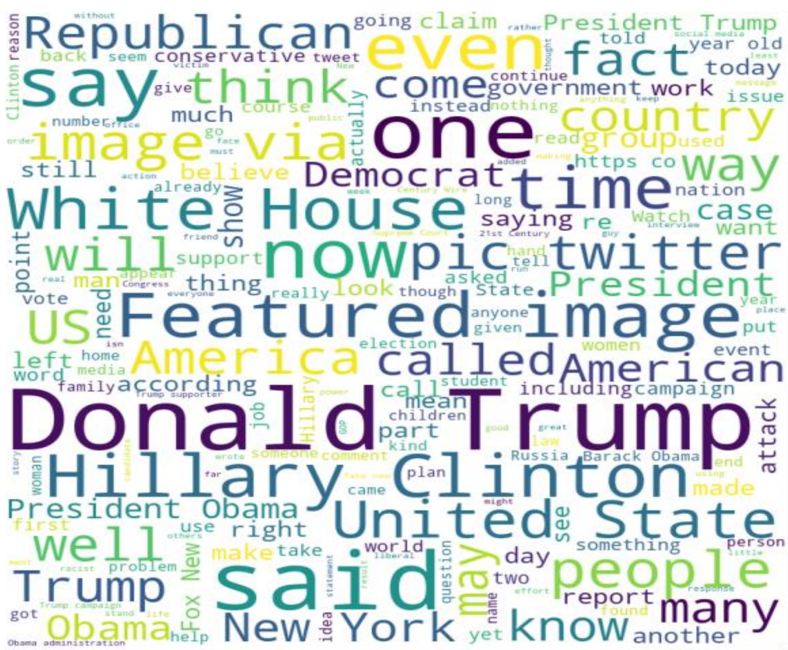

Figure 1. Word Cloud for Fake News

[1] https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

Figure 2. Word Cloud for True News

Second Dataset

A second less-biased news article dataset[2] (https://www.kaggle.com/c/fake-news/data) was also used in this research with the same preprocessing steps applied on the first dataset, for comparison purposes. This dataset includes the text of the news article along the target variable 'label' that marks the article as potentially unreliable (i.e., 1: unreliable and 0: reliable). There are 20,000 rows in this dataset (i.e., train.csv file).

Data Preprocessing and Preparation

Concatenation of true news dataset and fake news dataset is performed using pandas' data frame, with label 'class' added to distinguish fake (class=0) from true (class=1). News texts were then represented using bag of words features with using TF-IDF as weighting scheme. TF-IDF weight of a feature i in a document j is:

$$TF_{i,j} + (TF_{i,j} * \log(N/DF)) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where $TF_{i,j}$ is the frequency of the feature i in the text j and N indicates the number of news in the corpus. DF is the number of news that contain feature i. The effect of this is that features with zero IDF, i.e., that occur in all news (i.e., texts) of a training set will not be entirely ignored. TF is normalized using the sum of all TFs in the question or the post.

Analysis of first data set lead to the discovery that news articles with 'Reuters' tf-idf greater than 0.00 tend to be true news while the rest being fake news. Essentially this means the dataset is biased to build a more general fake news classifier. Therefore, additional processing steps are taken, i.e., removing the column 'Reuters' from pandas data frame. Updated datasets result in more reasonable supervised machine learning accuracy, as outlined in detail in the result section.

[2] https://www.kaggle.com/c/fake-news/data

Model Building and Description

Baseline Models

The baseline learning algorithms used are decision tree, naive bayes and neural network, with feature selection. *TfidfVectorizer* was used to extract term frequency inverse document frequency with English stop words removed, the result is filtered with condition that 0.25 < df < 0.75. Then resulting data is fed into the models to perform binary classification tasks. For feature selection, χ^2 is calculated and p value < 0.05 is chosen as relevant. For the neural network, two hidden layers with 5 nodes and 2 nodes respectively are included in the model, with the optimizer chosen to be either '*adam*' or '*lbfgs*' (it turns out, contrary to what is suggested by SciKitLearn, '*lbfgs*' actually performs faster and better than '*adam*').

Word Embedding-Based Long Short-Term Memory (LSTM) Approach

In the proposed approach, we apply Long Short-Term Memory (LSTM) algorithm with word embeddings. LSTM is a recurrent neural network (RNN). In LSTM, stored values are not changed when learned progress is made. Furthermore, RNNs allow forward and reverse connections between neurons while processing the data. Figure 3 shows the architecture of the proposed approach. The model consists of three layers, embedding (input layer), LSTM (hidden layer), and dense layer (output layer). The output of the embedding layer is the input to the LSTM layer. The weights for the embedding layer were initialized using Word2Vec words embeddings. We chose dimension 100 so each word will be converted to sequence of 100 vectors. Therefore, each word in the corpus was assigned to some sequence in the tokenizer. The advantage of using word embeddings is to capture the semantic relationship between words (i.e., learn the similarity between the words in the corpus). In the output layer (dense layer), we used the activation function "sigmoid". We used Tensorflow Keras library in Python to implement the model. Figures 4 and 5 summarize the models created for the first and second datasets respectively.

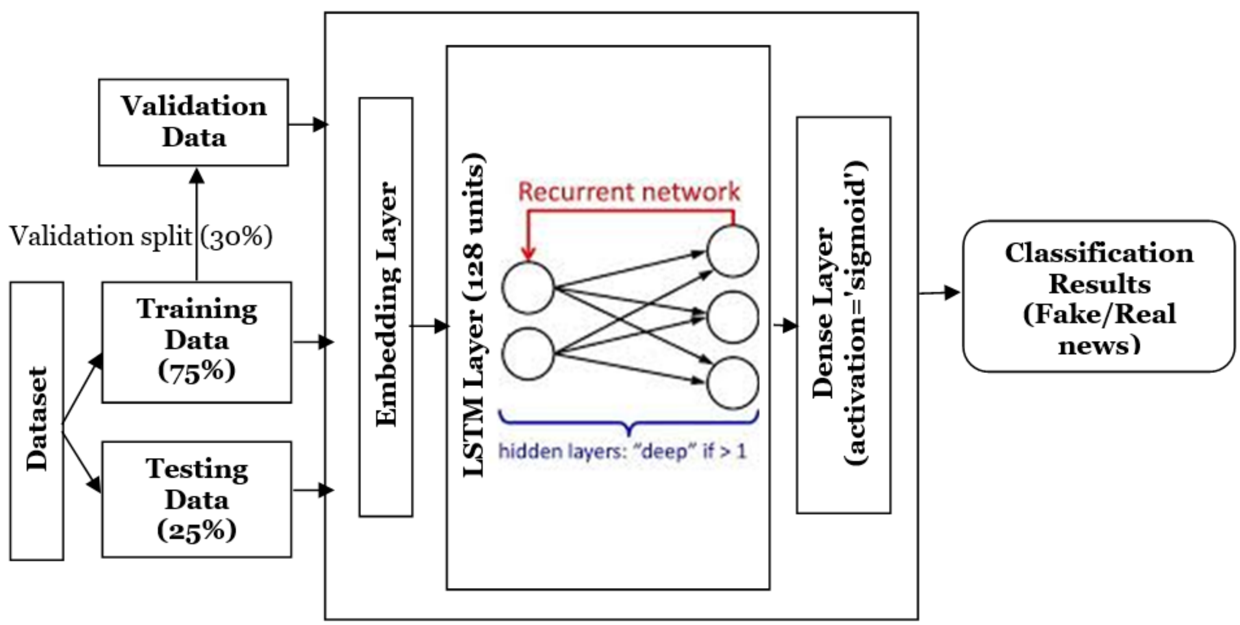

Figure 3. Architecture of the Proposed Approach

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         23187300

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 23,304,677
Trainable params: 117,377
Non-trainable params: 23,187,300
```

Figure 4. LSTM model summary using the first dataset

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1000, 100)         22694900

 lstm (LSTM)                 (None, 128)               117248

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 22,812,277
Trainable params: 117,377
Non-trainable params: 22,694,900
```

Figure 5. LSTM model summary using the second dataset

Model Evaluation

Datasets were splitted into training (0.75) and testing (0.25). Once training has been completed, the testing dataset can be used to evaluate the model. Using the trained model to predict the target 'class'/'label' from the testing dataset, the result can be evaluated using the accuracy measure. The accuracy metric measures the percentage of those correctly classified as positive or negative examples. The reason we utilized the accuracy measure is to facilitate comparison with existing models in pertaining literature since the existing papers did not provide the measures such as recall, precision, and F1 score.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Where

TP: The number of correctly predicted positive samples

TN: The number of correctly predicted negative samples

FP: The number of incorrectly predicted positive samples

FN: The number of incorrectly predicted negative samples

Results and Discussion

Table 2 shows the models' performance using the first and second datasets. The experiment results reported better performance of the proposed model compared with the other baseline models. Our model achieved a high accuracy of 99% and 96% in classifying the news articles using the first and second datasets respectively.

	First Dataset	Second Dataset
	Accuracy (%)	
Decision Tree	88	76
Naive Bayes	82	74
Neural Network	87	81
LSTM and Word Embedding	**99**	**96**

Table 2. Models' Performance

We then compared the performance of our suggested model with the existing models for the detection of fake news using the Kaggle news dataset. As shown in Table 3, our model can potentially outperform existing models for news articles. The table also shows that deep learning models that use embeddings outperform other regular machine learning models, which demonstrate that pre-trained word embedding models play a significant role in the fake news detection.

Model	**Accuracy**
Naive Bayes (Granik and Mesyura 2017)	74%
Random forest (Singh et al. 2021)	86%
Naive Bayes Algorithm with enriched corpora (Adiba et al. 2020)	92%
LinearSVC (Kaur et al. 2020)	95.9%
Linear Support Vector Machine (LSVM) (Ahmed et al. 2017)	92%
Recurrent Neural Network (Ruchansky et al. 2017)	89%
Multi-Layer Perceptron (MLP) classifiers (Kotteti et al. 2018)	44%
Models with word embeddings	
Our proposed model	**99% (first dataset), 96% (second dataset)**
GloVe-enabled deep convolutional-based approach (Kaliyar et al. 2020)	98.36%
Long Short-Term Memory (LSTM) (Vyas et al. 2021)	89.95%
Text and Image information based Convolutional Neural Network (TI-CNN) (Yang et al. 2018)	92%
Deep neural networks (convolutional neural networks) (O'Brien et al. 2018)	93.5%

Table 3. Comparison of our Proposed Model with Existing Models Using the Kaggle News dataset

Conclusion

With the exponential increasing use of social media, it becomes easier for people to consume news from social media rather than traditional news media. However, social media has also been used to spread fake news, which has strong negative impacts on individual users and broader society. In this research, we propose LSTM-based approach using word embeddings for fake news detection. To ensure the generalizability of our results, we utilized two datasets of news articles published on Kaggle website. Experiments results show that the proposed model outperform the baseline machine learning models with higher accuracy. We also compared our results with pertaining studies that used Kaggle news article datasets. Overall, our results revelated that pre-trained word embedding models play a significant role in the fake news detection.

As future work, we aim to evaluate our proposed model on other data sets for fake news detection like news tweets. Further, the efficiency of the proposed model in detecting misinformation in other domains like healthcare (i.e., for example misinformation about COVID-19) worth investigation.

REFERENCES

Adiba, F.I., Islam, T., Kaiser, M.S., Mahmud, M., and Rahman, M.A. 2020. "Effect of Corpora on Classification of Fake News Using Naive Bayes Classifier," *International Journal of Automation, Artificial Intelligence and Machine Learning* (1:1), pp. 80-92.

Ahmed, A.A.A., Aljabouh, A., Donepudi, P.K., and Choi, M.S. 2021. "Detecting Fake News Using Machine Learning: A Systematic Literature Review," *arXiv preprint arXiv:2102.04458*).

Ahmed, H. 2017. "Detecting Opinion Spam and Fake News Using N-Gram Analysis and Semantic Similarity."

Ahmed, H., Traore, I., and Saad, S. 2017. "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*: Springer, pp. 127-138.

Al-Ramahi, M., and Alsmadi, I. 2021. "Classifying Insincere Questions on Question Answering (Qa) Websites: Meta-Textual Features and Word Embedding," *Journal of Business Analytics* (4:1), pp. 55-66.

Al-Ramahi, M.A., and Alsmadi, I. 2020. "Using Data Analytics to Filter Insincere Posts from Online Social Networks. A Case Study: Quora Insincere Questions," *The 53rd Hawaii International Conference on System Sciences (HICSS-53)*, Hawaii.

Balwant, M.K. 2019. "Bidirectional Lstm Based on Pos Tags and Cnn Architecture for Fake News Detection," *2019 10th International conference on computing, communication and networking technologies (ICCCNT)*: IEEE, pp. 1-6.

Choudhary, A., and Arora, A. 2021. "Linguistic Feature Based Learning Model for Fake News Detection and Classification," *Expert Systems with Applications* (169), p. 114171.

Ghosh, S., and Shah, C. 2018. "Towards Automatic Fake News Classification," *Proceedings of the Association for Information Science and Technology* (55:1), pp. 805-807.

Granik, M., and Mesyura, V. 2017. "Fake News Detection Using Naive Bayes Classifier," *2017 IEEE first Ukraine conference on electrical and computer engineering (UKRCON)*: IEEE, pp. 900-903.

Islam, M.U., Hossain, M.M., and Kashem, M.A. "Covfake: A Word Embedding Coupled with Lstm Approach for Covid Related Fake News Detection," *International Journal of Computer Applications* (975), p. 8887.

Jadhav, S.S., and Thepade, S.D. 2019. "Fake News Identification and Classification Using Dssm and Improved Recurrent Neural Network Classifier," *Applied Artificial Intelligence* (33:12), pp. 1058-1068.

Kaliyar, R.K., Goswami, A., Narang, P., and Sinha, S. 2020. "Fndnet–a Deep Convolutional Neural Network for Fake News Detection," *Cognitive Systems Research* (61), pp. 32-44.

Kaur, S., Kumar, P., and Kumaraguru, P. 2020. "Automating Fake News Detection System Using Multi-Level Voting Model," *Soft Computing* (24:12), pp. 9049-9069.

Kesarwani, A., Chauhan, S.S., and Nair, A.R. 2020. "Fake News Detection on Social Media Using K-Nearest Neighbor Classifier," *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*: IEEE, pp. 1-4.

Kotteti, C.M.M., Dong, X., Li, N., and Qian, L. 2018. "Fake News Detection Enhancement with Data Imputation," *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*: IEEE, pp. 187-192.

Kurasinski, L. 2020. "Machine Learning Explainability in Text Classification for Fake News Detection." Malmö universitet/Teknik och samhälle.

Li, J., Ott, M., Cardie, C., and Hovy, E. 2014. "Towards a General Rule for Identifying Deceptive Opinion Spam," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1566-1576.

Meneses Silva, C.V., Silva Fontes, R., and Colaço Júnior, M. 2021. "Intelligent Fake News Detection: A Systematic Mapping," *Journal of applied security research* (16:2), pp. 168-189.

O'Brien, N., Latessa, S., Evangelopoulos, G., and Boix, X. 2018. "The Language of Fake News: Opening the Black-Box of Deep Learning Based Detectors,").

Ott, M., Choi, Y., Cardie, C., and Hancock, J.T. 2011. "Finding Deceptive Opinion Spam by Any Stretch of the Imagination," *arXiv preprint arXiv:1107.4557*).

Ruchansky, N., Seo, S., and Liu, Y. 2017. "Csi: A Hybrid Deep Model for Fake News Detection," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 797-806.

Sharma, R., Agarwal, V., Sharma, S., and Arya, M.S. 2021. "An Lstm-Based Fake News Detection System Using Word Embeddings-Based Feature Extraction," in *Ict Analysis and Applications*. Springer, pp. 247-255.

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. 2017. "Fake News Detection on Social Media: A Data Mining Perspective," *ACM SIGKDD explorations newsletter* (19:1), pp. 22-36.

Singh, V.K., Ghosh, I., and Sonagara, D. 2021. "Detecting Fake News Stories Via Multimodal Analysis," *Journal of the Association for Information Science and Technology* (72:1), pp. 3-17.

Vyas, P., Liu, J., and El-Gayar, O. 2021. "Fake News Detection on the Web: An Lstm-Based Approach,").

Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., and Yu, P.S. 2018. "Ti-Cnn: Convolutional Neural Networks for Fake News Detection," *arXiv preprint arXiv:1806.00749*).