Detecting Fake News with Machine Learning

Nagender Aneja[1] and Sandhya Aneja[2]

[1] Institute of Applied Data Analytics, Universiti Brunei Darussalam, Brunei Darussalam,
nagender.aneja@ubd.edu.bn,
[2] Faculty of Integrated Technologies, Universiti Brunei Darussalam, Brunei Darussalam
sandhya.aneja@ubd.edu.bn,
http://expert.ubd.edu.bn/sandhya.aneja

Abstract. Fake news is intentionally written to influence individuals and their belief system. Detection of fake news has become extremely important since it is impacting society and politics negatively. Most existing works have used supervised learning but given importance to the words used in the dataset. The approach may work well when the dataset is huge and covers a wide domain. However, getting the labeled dataset of fake news is a challenging problem. Additionally, the algorithms are trained after the news has already been disseminated. In contrast, this research gives importance to content-based prediction based on language statistical features. Our assumption of using language statistical features is relevant since the fake news is written to impact human psychology. A pattern in the language features can predict whether the news is fake or not. We extracted 43 features that include Parts of Speech and Sentiment Analysis and shown that AdaBoost gave accuracy and F-score close to 1 when using 43 features. Results also show that the top ten features instead of all 43 features give the accuracy of 0.85 and F-Score of 0.87.

Keywords: Fake news, Machine Learning, AdaBoost, Decision Tree, Naive Bayes; K-Nearest Neighbors; Stochastic Gradient Descent, Support Vector Machine

1 Introduction

One of the challenging problems for traditional news media and social media service providers in Natural Language Processing (NLP) is to detect fake news due to its social and political impact on individuals [1]. It is also important since the capability of humans to detect deceptive content is minimal, especially when the volume of false information is high. Although fake news has been

Citation: Aneja N., Aneja S. (2021) Detecting Fake News with Machine Learning. Conference Proceedings of ICDLAIR2019. ICDLAIR 2019. Lecture Notes in Networks and Systems, vol 175. Springer, Cham. https://doi.org/10.1007/978-3-030-67187-7_7

around from a long time as propaganda, however, it has grown exponentially in recent years. Privately owned websites and social media users/groups have amplified the distribution of fake news since anyone can create a website or social media page and claim as news media. Social media has advantages to sharing information informally, however, this feature has been misused by a few people or organizations to distribute unverified content. The content which is well documented but fake is being distributed for political or other malicious purposes. The objective of fake content writers is to influence beliefs and thus to impact users' decisions. Social media has become a place for campaigning misinformation that affects the credibility of the entire news ecosystem.

Another issue that prevents social media users from seeing both sides of the coin is the informational separation caused by filtration of information through news aggregators [2]. Newsfeed of a user is most likely to contain posts of his friends who have the same attitude, and thus belief of the user is influenced by such posts. While on the other hand, information about different point of view doesn't reach to a user. This is more a rational and social issue, wherein, the users may be algorithmically advised if the news feed of a user represents one view only.

Fake news is defined as a piece of news, which is stylistically written as real news but is entirely or partially false [3]. Undeutsch hypothesis also [4] states that a fake statement differs in writing style and quality from a true one. Recent techniques are based on content, however, the fundamental theories in social and forensic psychology have not played a significant role in these techniques. Research efforts have been to automate the detection of fake news so that a user is informed about the content even if his or her friends share it. Fully automatic detection is still a research topic, however, supervised machine approaches that identify patterns in the fake news are being explored.

2 Problem Statement

Identification of fake news is a binary classification problem since there are two classes, fake and real. Mathematically, the problem may be stated as follows.

Let $N = \{n_1, n_2, n_3, ..., n_M\}$ be a collection of M news items and $L = \{l_1, l_2, l_3, ..., l_M\}$ be their corresponding labels of news items such that label l_i of news item n_i is either 1 or 0 depending on if the news item n_i is fake or real.

We need a machine-learning algorithm that can predict the accurate label of news item $n_z \notin N$.

Most of the current approaches are based on a dictionary that is developed from news items $n_i \in N$. In other words, a dictionary $D = \{w_1, w_2, w_3, ..., w_K\}$ of K words is such that the all words of new items $words_{n_i} is \subset D \; \forall \; n_i \in N$ Thus, performance of the model for a news item $n_z \notin N$ will be based on similarity of n_z with N.

Instead of a dictionary-based approach, we want to evaluate if the language features can be used to detect fake news. The assumption may work since there is

clear intention to write fake news, and the content in fake news is written based on human psychology to influence social belief system. In this research, we extracted numerical features from all news items and used the machine learning binary classification algorithm to detect fake news. Since we don't give importance to the similarity of words within the dataset, this approach may generalize well on other datasets also. The following section describes the proposed algorithm.

3 Related Work

This section describes a survey of recent prior work published in the area of fake news detection.

Guacho et al. [5] proposed semi-supervised content-based method that uses tensor-based article embeddings to construct a k-nearest neighbor graph of news articles that captures similarity in a latent, embedding space. The authors showed 75:43% accuracy using 30% of labels of one dataset and 67:38% accuracy using 10% labels of another dataset. Additionally, the method attains 70:92% accuracy using only 2% labels of the dataset.

Oshikawa et al. [6] presented a review on natural language processing solutions for automatic fake news detection and developed a textual content-based method on multi-class fake news detection based on natural language processing.

Zhou et al. [7] investigated news content at various levels: lexicon-level, syntax-level, semantic-level and discourse-level. The authors observed that the current techniques capture non-latent characteristics e.g. word-level statistics based on Term Frequency-Inverse Document Frequency (TF-IDF) Pérez-Rosas et al. [8], n-gram distribution Pérez-Rosas et al. [8] and/or utilize Linguistic Inquiry and Word Count (LIWC) features Pennebaker et al. [9]. Recently neural networks have also been used using the latent characteristics within news content Volkova et al. [10], Wang et al. [11].

Gravanis et al. [12] exploited the use of linguistic-based features in combination with Machine Learning methods to detect news with deceptive content. The proposed features combined with ML algorithms obtained the accuracy of up to 95% with the AdaBoost as first in rank and SVM & Bagging algorithms to be next in ranking but without statistically significant difference. The authors used Linguistic features as proposed by Burgoon et al. [13], Newman et al. [14], Zhou et al. [15].

Reis et al. [16] proposed that recent work in fake news detection identify pattern after these have been disseminated. Thus, it is difficult to gauge the potential that supervised models trained from features proposed in recent studies can be used for detecting future fake news. The authors used Language Features, Lexical Features, Psycholinguistic Features, Semantic Features, and Subjectivity. The results have shown that Random Forest and XGBoost perform better.

4 Proposed Method

We propose two phase-process that includes extracting numerical features in the first phase and then using the numerical features to predict the label of the news item using machine learning classifiers in the second phase.

4.1 Datasets

To determine the feature set that can help to predict news items as fake or real, we consider two datasets, namely dataset of new items labeled fake and dataset of new items labeled real.

Fake news dataset was taken from Kaggle website [17] and Real news dataset was downloaded from Guardian website [18]. We only considered the text of news and ignored other metadata. The reason for using text is to find language style of authors in writing fake content vs. real content. Our final dataset included 12249 fake news items and 9725 real news items after considering only news items with the English Language.

4.2 Preprocessing and Features Extraction

In the preprocessing step, first, we cleaned the news items so that there is no special character and in case there is a special character we split the word at the special character, e.g., refugees/immigrants was split into words refugees and immigrants. After cleaning the news items, we tokenized each news items using the tokenizer function of the nltk library and filtered the stopwords using nltk corpus.

To extract features of all news items, we applied Parts of Speech (POS) pos tag and Vader sentiment of nltk on filtered words representing news items. Table 1 provides 43 features that were extracted from pos tagging and sentiment analysis in addition to counting unique words. The description of POS tags have been explained in Table 2.

Table 1. Features Set

POS Tags (39)	$, “”, ., :, CC, CD, DT, EX, FW, IN, JJ, JJR, JJS, MD, NN, NNP, NNPS, NNS, PDT, POS, PRP, PRP$, RB, RBR, RBS, RP, SYM, TO, UH, VB, VBD, VBG, VBN, VBP, VBZ, WDT, WP, WP$, WRB
Sentiment (3)	’positive’, ’neutral’, ’negative’
Miscellaneous (1)	’unique’

We applied sentiment analysis to extract three features namely number of positive words, negative words, and neutral words from filtered words after converting the words to its base form using lemmatization. The process of lemmatization converts a word to meaningful base form while still maintaining the

Table 2. Description of NLTK POS Tags used in the present dataset

#	POS Tag	Description	POS Tag	Description
1	$	dollar	“”	quotes
2	.	Dot	:	colon
3	CC	conjunction, coordinating	CD	numeral, cardinal
4	DT	determiner	EX	existential there
5	FW	foreign word	IN	preposition or conjunction, subordinating
6	JJ	adjective or numeral, ordinal	JJR	adjective, comparative
7	JJS	adjective, superlative	MD	modal auxiliary
8	NN	noun, common, singular or mass	NNP	noun, proper, singular
9	NNPS	noun, proper, plural	NNS	noun, common, plural
10	PDT	pre-determiner	POS	genitive marker
11	PRP	pronoun, personal	PRP$	pronoun, possessive
12	RB	adverb	RBR	adverb, comparative
13	RBS	adverb, superlative	RP	particle
14	SYM	symbol	TO	”to” as preposition or infinitive marker
15	UH	interjection	VB	verb, base form
16	VBD	verb, past tense	VBG	verb, present participle or gerund
17	VBN	verb, past participle	VBP	verb, present tense, not 3rd person singular
18	VBZ	verb, present tense, 3rd person singular	WDT	WH-determiner
19	WP	WH-pronoun	WP$	WH-pronoun, possessive
20	WRB	Wh-adverb		

context instead of stemming that removes a few characters from the end. For example, lemmatization of caring is care, while stemming of caring is car. The word set that we get after lemmatization is the words that represent a particular news item for Sentiment Analysis that gives the number of positive, neutral, negative words and miscellaneous feature that provides the unique number of words.

Finally, we divided all features of POS tags, Sentiment Features, and Number of Unique words with the total number of words so that the value of the feature lies between 0 to 1. This is in contrast to other published work in which features were divided by the number of sentences. Next section describes the application of Machine Learning Algorithms on the feature set.

4.3 Supervised Learning

In this research, we employed several supervised machine learning algorithms to model the fake and real news items accurately. We determined the best candidate algorithm from preliminary results and further optimized the algorithm to best model the data.

Data Exploration Table 3 provides the information that the dataset includes 12249 fake news and 9725 real news items. Total 43 features were extracted as explained in Table 1.

Table 3. Dataset

Fake news items	12249
Real news items	9725
Total News Items	21974
Features Extracted for each news item	43

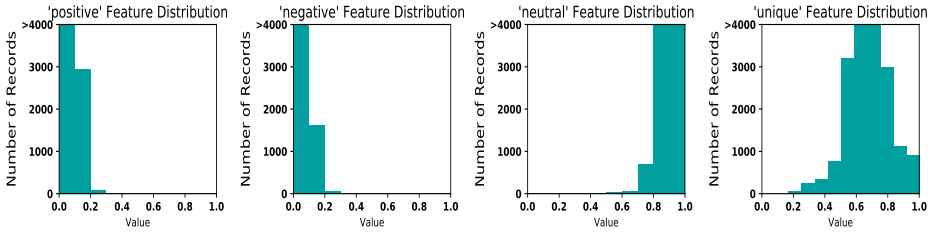

Fig. 1. Sentiment Feature Distribution before Normalization

Figure 1 describes Feature Distribution (before normalization) for sample features {Positive, Negative, Neutral, Unique words} wherein the values lies between 0 to 1. The features were normalized so that the mean value of each feature is 0, and the standard deviation is 1. Figure 2 shows Features Distribution for sample features {Positive, Negative, Neutral, Unique words} after normalization.

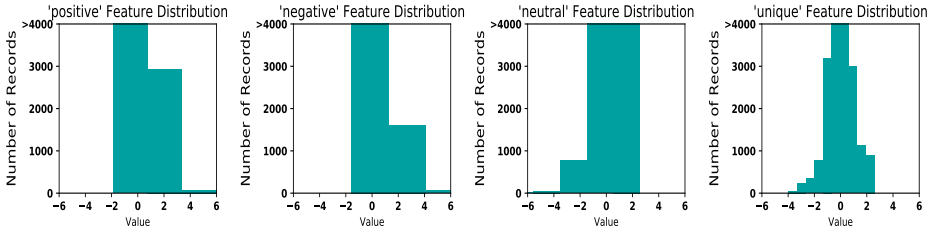

Fig. 2. Sentiment Feature Distribution after Normalization

The dataset of 21974 news items was divided into a training set and testing set using sklearn train_test_split function so that 80% of the data is used for

training and 20% of the data is used for testing. Thus, the training set included 17579 samples, and the testing set included 4395 samples.

Evaluating Model Performance We implemented Logistic Regression (LR), Stochastic Gradient Descent Classifier (SGDC), Support Vector Machines, K-Nearest Neighbors (KNeighbors), Gaussian Naive Bayes (GaussianNB), and Decision Trees in addition to Naive Predictor that always predict news item as not fake. To evaluate model performance, accuracy may be appropriate, however, predicting a piece of real news as fake may be a concern. Thus, we used a metric based on precision and recall.

Accuracy measures the correct output, whether correctly predicted fake or correctly predicted real from total news items from the test dataset. Precision, as shown in Equation 1, measures the proportion of news items that the system classified as fake were fake. Recall, as shown in Equation 2, tells as what proportion of fake news items were identified as fake. Thus,

$$Precsion = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

A model's ability to precisely predict fake is more critical than the model's ability to recall. We may use F-beta score ($\beta = 0.5$), as shown in Equation 3, as a metric that considers both precision and recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{3}$$

Naive Predictor The purpose of Naive Predictor is to show what a base model without intelligence would perform. A model that predicts all news items as Fake gives Accuracy score of 0.5574 and F-score of 0.6116.

Supervised Learning Models We implemented the following six machine learning models.

1. Ada Boost Classifier
2. Decision Trees Classifier
3. Gaussian Naive Bayes (GaussianNB)
4. K-Nearest Neighbors (KNeighbors)
5. Stochastic Gradient Descent Classifier (SGDC)
6. Support Vector Machine

AdaBoost trains multiple weak learners and combines the weak learners. The algorithm is very fast and able to boost performance. The weakness of AdaBoost is that it is sensitive to noise or outliers since a week learner may increase contributions of noise or outliers.

Decision Trees are non-parametric methods used for classification and regression. It is a tree of decisions that predicts the target variable based on decision rules. The deeper the curve, more complex decision rules and better fitting. A decision tree is a good candidate algorithm since there should be few features that will contribute more to predict and thus a decision tree is an easy tool for this type of problem. It also has no significant impact on outliers.

Naive Bayes is a good algorithm for working with text classification. The relative simplicity of the algorithm and the independent features assumption of Naive Bayes make it a good candidate for classifying texts. Further, Naive Bayes works best when training data set and features (dimensions) is small. In case of a huge feature list, the model may not give better accuracy, because the likelihood would be distributed and may not follow the Gaussian or other distribution. Naive Bayes works best if the features are independent of each other, which looks like the case when we plotted the data using a scatter diagram.

K-Nearest Neighbors (KNeighbors) provides functionality for the supervised and unsupervised model. Supervised nearest neighbors can be used for classification and regression problems. KNN finds a predetermined number of samples closest in the distance to a point to predict the label. The samples can be k in numbers or based on the local density of points (radius-based neighbor learning). Although, it may take time the KNN algorithm is simple to visualize and can easily find similarity/patterns in the data.

Stochastic Gradient Descent Classifier (SGDC) is generally useful when number of features and samples are large. SGDClassifier and SGDRegressor fit linear models for classification and regression using different loss functions. with log loss SGDClassifier fits Logistic Regression (LR) and with hingle loss it fits a Linear Support Vector Machine. SGD classifier implements regularized linear models using SGD (GD - full dataset; SGD - 1 sample to update weight; MBGD - mini batch to update weights).

Support Vector Machines SVM is used for classification, regression, and outliers detection and is effective in high dimension spaces and even when the number of dimensions is greater than samples. Since it uses a subset of training points (supporting vectors) in decision function so is memory efficient. It is versatile since different Kernel Functions can be used in the decision function.

The initial performance of the above models was tested using default configurations based on three training sets considering (i) 1% of training data, (ii) 10% of training data, and (i) 100% of training data.

Figure 3 shows results for the time taken, accuracy score, F-Score for both phases, namely Training and Testing. Training time at 100% of the dataset was highest for Support Vector Machines, followed by Ada Boost Classifier. However, the testing time of K-Nearest Neighbors was observed highest. Most of the algorithms performed well and gave close to 1 accuracy and F-score when 100% of the training data used.

Table 4 shows that results close to 1 but AdaBoost performed well in case of Test Accuracy and Test F-Score. The default base estimator of AdaBoost Classifier is DecisionTreeClassifier with max_depth as 1.

Fig. 3. Initial Results with 1%, 10%, and 100% of training data

Table 4. Initial Performance with 100% of training data

Algorithm	Test Accuracy	Test F-Score
AdaBoostClassifier	**0.9989**	**0.9993**
DecisionTreeClassifier	0.9980	0.9983
GaussianNB	0.9902	0.9867
KNeighborsClassifier	0.9747	0.9822
SGDClassifier	0.9859	0.9851
SVC	0.9966	0.9968

Adaboost (Adaptive Boosting) combines classifiers with poor performance, also known as, weak learners, into a bigger classifier with much higher performance. Adaboost starts with a week learner that classifies some points correctly but misclassify few. This week learner is added into the list and train another week learner that gives more weight to classify the previous mis-classify points and also may mis-classify few other points. This week learner is saved and the algorithm continues to train another week learner that gives more weights to classify the misclassified points of the previous learner. This approach is continued up to a threshold, and the final model is a combination of the different week learners. Thus, each classifier focus on the mistakes of the previous classifier. AdaBoost is particularly useful when it is easy to create a simple classifier, e.g., in this case, we know that few features can predict the style of news content to predict the correct label. Adaboost can create simple classifiers and optimally

combine them. Once a model has been trained, it can easily predict whether the news is fake or not.

Results also indicate the performance of AdaBoost is better than others even when reduced training data is used. The F-score increases with an increase in training sample size and is maximum when 100% training data is used. The training time of AdaBoost is higher when 100% training data is used but should be excellent due to negligible prediction time and higher F-Score.

Improving Results by Model Tuning Since AdaBoost performed well in the initial results, so we fine-tuned AdaBoost using GridSearchCV module of sklearn. We considered the following parameters for GridSearch.

1. *n_estimators*: 50, 100, 150, 175, 200
2. *base_estimator*: Decision Tree Classifier with max depth as 1, 2, and 3
3. scorer: F-Score with beta 0.5
4. cv: 5

The default value of n_estimators is 50 that defines a maximum number of estimators at which boosting is terminated. In case of a perfect fit, the learning procedure is stopped early. The default value of base estimator from which the boosted ensemble is built Decision Tree Classifier with max depth 1. The cv stands for cross-validation that defines cross-validation splitting strategy. The default value of cv is 3 specifies the number of folds in Stratified KFold.

Grid Search provided the best parameters with max depth 3 and the number of estimators as 175. AdaBoost with optimized parameters gave accuracy and F-Score equal to 1.0000 that was an improvement from the accuracy of 0.9989 and F-score of 0.9993. Figure 4 shows confusion matrix for AdaBoost with best tuned parameters.

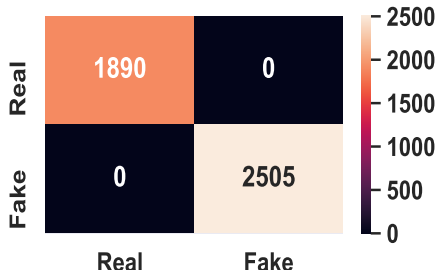

Fig. 4. Confusion Matrix based on AdaBoost Test results

Feature Importance The section describes the importance of particular features for the predictive power to help understand the crucial features for determination of fake news. Figure 5 shows the most crucial top ten features are NN

(noun, common, singular or mass); CD (numeral, cardinal); VBP (verb, present tense, not 3rd person singular); VBG(verb, present participle or gerund); positive (positive sentiment); NNP(noun, proper, singular); JJ(adjective or numeral, ordinal); IN(preposition or conjunction, subordinating); VBN(verb, past participle); and unique (unique words). These top ten features cover around 60% of features domain.

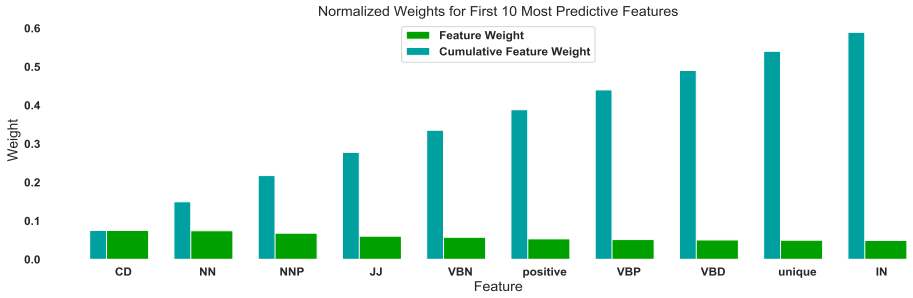

Fig. 5. Importance of Features

Additionally, to test the impact of top features, we again trained the model only on these essential ten features. AdaBoost with the best classifier provided accuracy of 0.8578 and F-Score of 0.8753 in comparison to the accuracy of 1.0000 and F-score on the full feature set. Thus, we can say that features NN (noun, common, singular or mass); CD (numeral, cardinal); VBP (verb, present tense, not 3rd person singular); VBG(verb, present participle or gerund); positive (positive sentiment); NNP(noun, proper, singular); JJ(adjective or numeral, ordinal); IN(preposition or conjunction, subordinating); VBN(verb, past participle); and unique (unique words) are highly essential to predict fake news.

5 Conclusion and Future Work

This research implemented AdaBoost classifier; DecisionTreeClassifier; GaussianNB; KNeighborsClassifier; SGDClassifier; and SVC to predict whether a piece of particular news is fake or real. Results show that AdaBoost Classifier with base estimator as Decision Tree of maximum depth 3 and 175 estimators performs best and provides accuracy close to 1 when 43 features were considered. Features NN, CD, VBP, VBG, positive, NNP, JJ, IN, VBN, and unique were found top predictive features that provided accuracy of 0.85 and F-score of 0.87. In future work, we will implement this algorithm on other datasets.

References

[1] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. The science of fake news. *Science*, 359(6380):1094–1096, 2018.

[2] Denis Zhuk, Arsenii Tretiakov, and Andrey Gordeichuk. Methods to Identify Fake News in Social Media Using Machine Learning. In *Proceedings of the 22st Conference of Open Innovations Association FRUCT*, page 59. FRUCT Oy, 2018.

[3] AP Sukhodolov. The phenomenon of "fake news" in the modern media space. *EURASIAN COOPERATION: HUMANITIES ASPECTS*, page 36, 2017.

[4] Udo Undeutsch. The development of statement reality analysis. In *Credibility assessment*, pages 101–119. Springer, 1989.

[5] Gisel Bastidas Guacho, Sara Abdali, Neil Shah, and Evangelos E Papalexakis. Semi-supervised content-based detection of misinformation via tensor embeddings. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 322–325. IEEE, 2018.

[6] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*, 2018.

[7] Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. Fake news early detection: A theory-driven model. *arXiv preprint arXiv:1904.11679*, 2019.

[8] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[9] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[10] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, 2017.

[11] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 849–857. ACM, 2018.

[12] Georgios Gravanis, Athena Vakali, Konstantinos Diamantaras, and Panagiotis Karadais. Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, 2019.

[13] Judee K Burgoon, J Pete Blair, Tiantian Qin, and Jay F Nunamaker. Detecting deception through linguistic analysis. In *International Conference on Intelligence and Security Informatics*, pages 91–101. Springer, 2003.

[14] Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675, 2003.

[15] Lina Zhou, Judee K Burgoon, Jay F Nunamaker, and Doug Twitchell. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group decision and negotiation*, 13(1):81–106, 2004.

[16] Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, Fabrício Benevenuto, and Erik Cambria. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81, 2019.

[17] Megan Risdal. Getting Real about Fake News. https://www.kaggle.com/mrisdal/fake-news, 2017. Online.

[18] Guardian News and Media Limited. The Guardian Open Platform. http://open-platform.theguardian.com/, 2017. Online.