Satire or Fake news? Machine Learning Based Approaches to Resolve the Dilemma

Protik Bose Pranto
ppranto@asu.com
Arizona State University
Arizona, USA

ABSTRACT

Fake news has become much more common in recent years due to the widespread availability of the internet and social media platforms. The damages that can be done by the rapid propagation of fake news in various sectors like politics and finance have drawn the attention of the research community to automatically identify fake news using linguistic analysis. However, due to the nature of satirical news, it can sometimes be challenging to separate it from fake news. Although several research studies have been conducted on identifying fake news, there has not been much work on satire news, particularly on detecting between fake and satirical news. To overcome this limitation, in this project, I examined nine widely used traditional machine learning models and three transformer-based traditional models (BERT, XLM-RoBERTa, DistilBERT) to see whether they can distinguish effectively between fake and satirical news. SVM performs better on a small dataset when text preprocessing and stemming are used. However, after text augmentation, the transformer-based model outperforms all other models, achieving 0.98 accuracy. So, by collecting a large amount of data on various topics and time periods, it is possible to effectively distinguish between fake and satirical news.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; • **Security and privacy** → *Human and societal aspects of security and privacy*; • **Human-centered computing** → Human computer interaction (HCI).

KEYWORDS

Misinformation, Fake News, Sarcasm, Satire, Machine Learning, Data Mining, Classification

1 INTRODUCTION

Fake news can be defined as a type of yellow journalism or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcast news media or online social media[22]. In recent years, the incidence of fake news has significantly expanded due to the rapid development of digitization and the rise of social media. For this reason, computer scientists have recently shown a great deal of interest in it. Numerous research explains how to identify false information in online content. On the other hand, irony and satire have received less attention as elements of natural human communication[8]. Nevertheless, it is extremely challenging to distinguish between satire and fake news [23]. Their basic motives are distinct. Satire uses fiction or humor to highlight a greater social or political truth. It only works if the audience is aware that it is fabricated. Whereas fake news operates under the guise of credible journalism to convince the audience of a falsehood, typically for political or financial advantage. It only works when the recipient is unaware of the lie. Therefore, satire plays with its audience, whereas fake news preys on it [3].

Some scholars argued that satire should not be included in the new definition of *Fake news* since it is *unlikely to be misconstrued as factual* and is not intended to inform audiences [2]. Others, however, think it should be included because, despite being legally protected speech, it could be misinterpreted as the truth [20]. For example, in 2017, a satire site run by a hoaxer named Christopher Blair said he was sorry because his story was *too real*. This was because many people did not realize it was a joke [12]. But the motivation and the targeted audience of satire and fake news are different, there will be differences in the storytelling approach while propagating these different types of articles [8]. This gives rise to the challenge of classifying fake news versus satire based on the content of a story.

2 RELATED WORK

Existing literature in the field of sarcasm detection comes from several disciplines, including linguistics, psychology, social science, and more recently, computer science. But their goals are different. For example, studies in psychology and the social sciences tend to focus on *why* and *when* questions, like *When and why do people use sarcasm, share or belief in misinformation*? On the other hand, studies in linguistics and computer science, tend to focus on *how* questions.

Over the years, numerous kinds of research have been done on the characteristics of fake news and its detection. Conroy et al. described three categories of fake news: Serious Fabrications, Large-Scale Hoaxes, and Humorous Fakes [27]. They define fake news as purposely and verifiably misleading news articles that could mislead readers [2]. A survey has found that Northerners are more inclined to find sarcasm amusing, and among them, men are more likely to self-identify as sarcastic [24]. According to *McClennen*, young people are using satire in entertainment media as a kind of political education and awareness [30].

For the automatic detection of fake news, various traditional machine learning-based algorithms have been devised by using different linguistic-based features such as total words, characters per

Conference'22, February 2022, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

word, frequencies of large words [29], n-grams, bag-of-words, parts-of-speech (POS) tagging [13], Probabilistic Context-Free Grammars (PCFG) [7], and bi-gram TF-IDF [25]. There have also been other studies that used deep learning models to detect fake news using different techniques such as hybrid convolutional neural network model [31], LSTM [26], bi-modal variational auto-encoder [19], node2vec [15], analyzing the relationship between the headline and the content of the news [17], hybrid architecture connecting BERT with RNN [21].

Horne et. al. claimed that algorithms may struggle to distinguish between satire and fake news because fake news contains many textual cues that make it resemble satirical news more than real news [16]. Using statistical data analysis, they aimed to determine which news stories are real, fake, or satire. *Golbeck et al.* also studied whether there are differences in the language of fake news and satirical articles on the same topic by following a word-based classification approach [14]. In addition, there is also some research interest in telling satirical news apart from real news [1, 5, 9, 10]. However, none of them conducted a thorough exploration of machine learning models for fake news detection and comparison, which was missing in previous research.

SVM performs better on small datasets in general. However, I hypothesize that traditional machine learning algorithms, when combined with stop words elimination, stemming, and text augmentation can outperform the baseline solution. Even after text augmentation, transformer-based pre-trained ML models can outperform all other ML models.

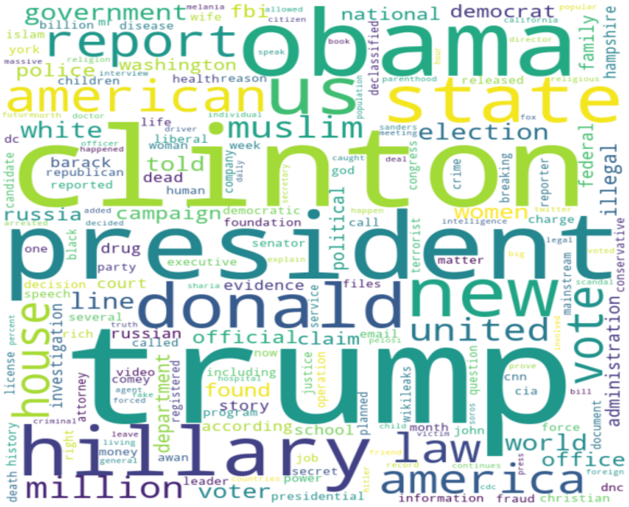

Figure 1: Most used words in the fake news content

3 DATASET

In this project, I am using the Fake News vs. Satire corpus [14] which contains 283 fake news articles and 203 satirical stories focused on American politics, posted between January 2016 and October 2017. The title, a link, and the full-text ID are provided for each article. For fake news stories, a rebutting article is also provided that disproves the premise of the original story achieving an accuracy of 79.1% with a ROC AUC of 0.880.

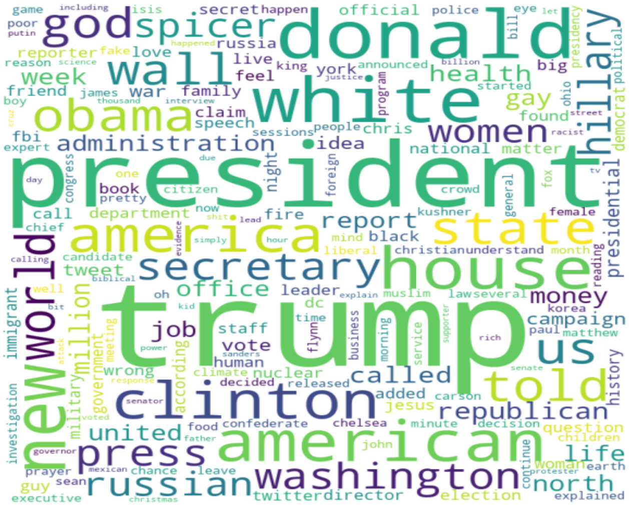

Figure 2: Most used words in the satire news content

4 METHODS

4.1 Data Preprocessing

Before feeding into the models, the raw text of the news required some preprocessing. I first eliminated unnecessary HTML links and URL addresses. The next step was to remove non-printable, non-English characters, punctuations, stop-words, and digits. I split every text by white space and remove suffices from words by stemming them. Finally, I rejoined the word tokens by white space to present our clean text corpus which had been tokenized later for feeding into the models. To execute all of the preprocessing, I used NLTK, a Natural Language Processing tool [4].

4.2 Data Analysis

Before evaluating the model, I attempted to determine which words the classifier would use to distinguish between satire and fake news. I used RAKE-NLTK to extract keywords from both fake and satire corpus.

Figure 1 and 2 shows the most used words in each class. As all the data in the dataset are politics related, that is why the words are also political-related. But most of the words are proper nouns e.g. Obama, Trump, etc.

Table 1: Top 4 keywords in each cluster (Fake Corpus)

Cluster no.	Main Keywords
1	Muslim, Islam, Women, Trump
2	Government, Trump, State, Obama
3	Hillary, Campaign, Women, President
4	Pelosi, Black, Russia, Senator
5	Vote, Election, Democrat, New
6	Melania, White, Trump, Obama

To gain a better understanding of the data, I separated the fake and satirical texts and used k-means clustering to discern what types of news are present. To determine the optimal number of clusters, I used the elbow method. The elbow method performs k-means clustering on the corpus for a range of k values (in my case, 2 to 10) and then computes the sum of square distances from each point to its assigned center for all clusters and finds the point of inflection from those distances that indicates the optimal cluster number for each value of k. Despite the interpretation of a line plot of the sum of squared distances for both the fake and satire corpus as straight lines with no elbow point, I assume the optimal cluster number for the fake corpus is 6 and for the satire corpus it is 7. Then, I extracted keywords from both the fake and satire corpora using the previous RAKE-NLTK. Table 1 and 2 shows the top 4 keywords in each cluster.

Table 2: Top 4 keywords in each cluster (Satire Corpus)

Cluster no.	Main Keywords
1	Ted, Cruz, Gay, Statue
2	Trump, American, Women, Health
3	Korea, Kim, North, Missile
4	President, Trump, Tweet, Wall
5	Hillary, Clinton, Secretary, Foundation
6	Moon, Fire, Trump, Ohio
7	White, Spicer, Cancer, House

The keywords in the clusters indicate that the majority of the news is about various political figures and their activities during the time period when the data was collected. Although the last two clusters from the satire corpus do not appear to be completely relevant, they could be the result of an incorrect cluster number assumption. However, the majority of the news is about the election, Trump, Korea, and various politicians' activities.

4.3 Data Augmentation

Due to the small dataset, there is a possibility that the models will underperform and that they will overfit. To tackle this problem, I used text augmentation to increase the training set and improve the model's performance. In the natural language processing (NLP) field, it is hard to augment text due to the high complexity of language. Also, I need to make sure that the semantic meaning of the augmented data stays the same, which means that satire remains satire and fake news remains fake, even after text augmentation. This is why I applied Back Translation (translating the text data to some language and then translating it back to the original language). This allows for the generation of textual data with unique words while preserving the context of the textual data.

Table 3: Original and Joined (Original+Augmented) Dataset Details

Dataset	Total Data	Fake	Satire	Type
Actual Data	486	283	203	Mild Imbalanced
Actual + Augmented Data	1068	534	534	Balanced

I used the Google Translate API provided by the Python library in five different languages (French, German, Spanish, Chinese, and Japanese). So, from a single news article, I receive five additional news articles. But, it is possible to receive the same back-translated news output as the initial text. I checked the similarity score between the freshly created augmented text and the input text for this reason. To assess the similarity score, I use the *term frequency-inverse document frequency (TF-IDF)* to determine the frequency of the words in the texts. I then use *cosine similarity* from *sklearn* to calculate the similarity, which ranges between 0 and 1. After applying a criterion of 0.6, I generated a total of 582 data, of which 251 are fake and 331 are satirical. After integrating them with the actual dataset, I obtained a total of 1068 data, 534 of which were marked as fake and the remaining 534 as satire. Table 3 represents all the information about the original dataset (Fake vs Satire Corpus) and joined dataset (Original Dataset + Augmented Dataset).

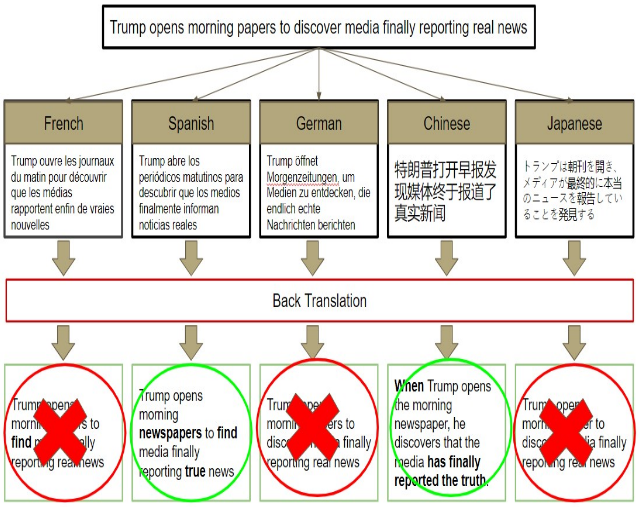
Trump opens morning papers to discover media finally reporting real news
French
Spanish
German
Chinese
Japanese
Trump ouvre les journaux du matin pour découvrir que les médias rapportent enfin de vraies nouvelles
Trump abre los periódicos matutinos para descubrir que los medios finalmente informan noticias reales
Trump öffnet Morgenzeitungen, um Medien zu entdecken, die endlich echte Nachrichten berichten
特朗普打开早报发现媒体终于报道了真实新闻
トランプは朝刊を開き、メディアが最終的に本当のニュースを報告していることを発見する
Back Translation
Trump opens morning papers to find media finally reporting real news
Trump opens morning newspapers to find media finally reporting true news
Trump opens morning papers to discover media finally reporting real news
When Trump opens the morning newspaper, he discovers that the media has finally reported the truth.
Trump opens morning paper to discover media finally reporting real news

Figure 3: Examples of Back-Translation. Some translated outputs are removed due to high similarity scores.

4.4 Experimental Setup

This is a classification problem since I need to determine whether a text is fake or satirical. I organized the entire experiment into steps. Initially, I conducted experiments using the actual data set (Fake News vs. Satire corpus). Since transformer-based deep learning models such as BERT, RoBERTa, etc. do not perform better on small datasets, I only experimented with widely used nine traditional machine learning models to discern how accurately they can detect fake or satirical news from the actual dataset. The models are KNN, Logistic Regression, Random Forest, Support Vector Machine, Naive Bayes, Decision Tree, XGBoost, AdaBoost, and Passive Aggressive Classifier. As I have only 486 data, I split the whole dataset into 90% training data and 10% test data to evaluate the models.

Table 4: Performance of traditional machine learning models on actual data

Model Name	Accuracy	Precision	Recall	F1 score
KNN	0.82	0.81	0.78	0.79
Random Forest	0.69	0.68	0.62	0.62
Logistic Regression	0.82	0.80	0.80	0.80
Support Vector Machine	**0.86**	0.85	**0.84**	**0.84**
Multinomial Naive Bayes	0.84	**0.86**	0.79	0.81
Decision Tree	0.65	0.62	0.61	0.61
XGBoost	0.69	0.67	0.64	0.65
Adaboost	0.69	0.67	0.64	0.65
Passive Aggressive	0.84	0.83	0.81	0.82

In the next step, I experimented with the joined dataset (Actual data + Augmented Data). As the dataset has increased in size, neural networks are expected to perform better than the traditional machine learning models. I used three Transformer-based pre-trained models to perform experiments on our dataset and they are BERT [11], XLM-RoBERTa [6], and DistilBERT [28]. These models, being pre-trained, perform considerably better than shallow neural networks such as LSTM, Bi-LSTM on a relatively small dataset [18]. These are all bi-directional transformers which means they are able to capture context from both right and left. I also used the previous nine traditional machine learning models. As this dataset is relatively larger than the actual dataset, I split the dataset into 80% training data and 20% test data to evaluate the models.

Due to the strong performance in various text classification tasks, I use the term frequency-inverse document frequency (TF-IDF) for the traditional machine learning models to evaluate how important a word is in the corpus. For the transformer models, the preprocessed input is passed to the Tokenizers of the corresponding models, such as BertTokenizerFast, XLMRobertaTokenizerFast, etc. The tokens from the Tokenizers are then passed to the models which return deep representations of the input texts. The deep representation is a 128-dimensional vector. I also used AdamW as the model optimizer, cross-entropy loss as the loss function, batch size 8, and 0.01 as the learning rate.

Table 5: Performance of Machine learning models (Actual + Augmented Data)

Model Name	Accuracy	Precision	Recall	F1 score
KNN	0.93	0.94	0.92	0.93
Random Forest	0.88	0.87	0.87	0.87
Logistic Regression	0.91	0.90	0.90	0.90
Support Vector Machine	0.90	0.90	0.88	0.89
Multinomial Naive Bayes	0.93	0.92	0.92	0.92
Decision Tree	0.73	0.72	0.72	0.72
XGBoost	0.82	0.81	0.82	0.81
Adaboost	0.85	0.84	0.84	0.84
Passive Aggressive	0.90	0.90	0.88	0.89
BERT	0.96	0.94	**0.98**	0.96
XLM-RoBERTa	**0.97**	**0.96**	**0.98**	**0.97**
DistilBERT	0.94	0.91	**0.98**	0.95

All the parameters for the traditional ML models are listed in table 6, and the remaining parameters are set as default as it is in the Scikit-learn library. I performed a 10-fold cross-validation for all the steps on the training data to better use this data, and to test the effectiveness of the machine learning models. I also applied the *GridSearchCV* technique to fine-tune and get the optimal parameter values for the models. I used the simpletransformers implementation for the models, which in turn uses the reputed open-source Transformers library hugging face. I performed experiments on NVIDIA Tesla K80 GPU provided by Google Colab. Finally, the test data is used to print the final evaluation after conducting the training using optimal parameters.

5 RESULTS

As this is a classification problem, the evaluation metric that is used in this project is accuracy (A), precision (P), recall (R), and F1 score. Accuracy is used to describe how the model performs across all classes. Recall also gives a measure of how accurately our model is able to identify the relevant data. Precision can be seen as a measure of quality, as it describes how many detected items are truly relevant. And, the F1 score is defined as the harmonic mean of precision and recall. In general, accuracy is used in a balanced dataset to measure a model's performance, while the F1 score is used in an imbalanced dataset.

For precision, recall, and F1-score, I considered the macro-average of both classes. In macro-average, all the individual classes are treated equally and the arithmetic mean of individual classes' scores is calculated. Here, table 4 shows the results of several machine learning models on the original data, whereas table 5 presents the results on the joined dataset.

In table 4, it can be seen that in terms of accuracy, recall, and F1-score, SVM is working better among all the machine learning models having an accuracy of 0.86 which is better than the baseline accuracy. Since the original dataset is mildly imbalanced, still SVM

Table 6: Optimal parameters for machine learning models

Model Name	Optimal Features (Actual Data)	Optimal Features (Actual Data + Augmented Data)
KNN	’metric’: ’cosine’, ’n-neighbors’: 12, ’weights’: ’distance’	’metric’: ’cosine’, ’n-neighbors’: 3, ’weights’: ’distance’
Random Forest	’criterion’: ’gini’, ’n-estimators’: 150	’criterion’: ’entropy’, ’n-estimators’: 150
Logistic Regression	’C’: 1000.0, ’max-iter’: 200, ’penalty’: ’l2’, ’solver’: ’saga’	’C’: 1000.0, ’max-iter’: 50, ’penalty’: ’l2’, ’solver’: ’sag’
Support Vector Machine	’C’: 100, ’gamma’: 0.01, ’kernel’: ’rbf’	’C’: 100, ’gamma’: 0.01, ’kernel’: ’rbf’
Multinomial Naive Bayes	’alpha’: 0.1	’alpha’: 0.001
Decision Tree	’ccp-alpha’: 0.01, ’criterion’: ’entropy’, ’max-depth’: 77, ’max-features’: ’auto’	’ccp-alpha’: 0.001, ’criterion’: ’entropy’, ’max-depth’: 94, ’max-features’: ’sqrt’
XGBoost	’eta’: 0.1, ’gamma’: 0.001, ’sampling-method’: ’uniform’, ’subsample’: 0.8	’eta’: 0.1, ’gamma’: 0.001, ’sampling-method’: ’uniform’, ’subsample’: 0.9
Adaboost	’algorithm’: ’SAMME’, ’learning-rate’: 0.5, ’n-estimators’: 500	’algorithm’: ’SAMME.R’, ’learning-rate’: 0.5, ’n-estimators’: 500
Passive Aggressive Classifier	’C’: 10, ’max-iter’: 500	’C’: 1, ’max-iter’: 2000

outperforms any other traditional ML model in terms of F1 score. However, in table 5, where transformer-based models and nine traditional models have been tested, XLM-RoBERTa outperforms all the other ML models with an accuracy of 0.97. All of the models’ performance improved after text augmentation, however in this instance, KNN and Naive Bayes surpassed all other traditional models.

6 DISCUSSION

As the original dataset is relatively small, SVM and Passive Aggressive Classifiers are expected to perform better. It is also evident in the evaluation result. In contrast to tree-based machine learning models, KNN and Multimodal Naive Bayes are also performing better. Cross-validation, according to Golbeck et al., improves the performance of Naive Bayes achieving 79.1% accuracy on this dataset. Das et. al. found that after removing stop words and digits, Naive Bayes can achieve 81% accuracy. Additionally, deleting stop words, URLs, and numbers, using stemming and GridsearchCV can result in better accuracy, in my case this is 86%.

Although the joined dataset is not large enough, transformer-based models are performing well. Even their recall score is very high, indicating that they can accurately identify relevant data. In general, RoBERTa has much more parameters and is supposed to perform better than BERT, which can be seen from the result. Also, the distilBERT is comparatively lightweight and performs slightly worse than BERT, which can also be seen. KNN and Naive Bayes outperform SVM and Passive Aggressive Classifiers because the joined dataset is considerably larger than the original dataset. However, tree-based classifiers, particularly decision trees, continue to perform badly. This may occur because the documents are relatively independent of one another. The data analysis reveals that although the clusters share some common keywords, their respective contexts are distinct. With respect to these independent behaviors, the tree-based approach struggles to perform better.

Although the models perform better when preprocessing and text augmentation are used, there are certain limitations. The high accuracy of the models may be due to the threshold score. Tweaking this score may result in changes to the accuracy score. Furthermore, all of the data is in text format, although in the real world, fake, and satirical news is delivered in both textual and visual formats. Another issue is that all of the data is about politics in a specific time period. As a result, for future study, a large amount of text and image-based data from various topics and time periods can be explored.

7 CONCLUSION

I evaluate the effectiveness of nine traditional machine learning and three transformer-based pre-trained models (BERT, XLM-RoBERTa, DistilBERT) in order to differentiate between satire and fake news. SVM is performing better in terms of accuracy, precision, and F1-score than any other traditional machine learning algorithms on a small dataset. But after the text augmentation, KNN, Naive Bayes, and all the transformer models are performing better which means that feeding a large amount of data into a pre-trained model can bring up good results and can be used in future research. After conducting a K-means clustering, it can be seen that most of the news is about the election, Trump, Korea, and various politicians’ activities. So, by collecting a large amount of data on various topics and time periods, it is possible to effectively distinguish between fake and satirical news.

REFERENCES

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy* 1, 1 (2018), e9.

[2] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* 31, 2 (2017), 211–36.

[3] Scott Anderson. 2020. Satire vs. fake news: University of Toronto Magazine. https://magazine.utoronto.ca/people/alumni-donors/satire-vs-fake-news-aaron-hagey-mackay/

[4] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O’Reilly Media, Inc.".

[5] Clint Burfoot and Timothy Baldwin. 2009. Automatic satire detection: Are you having a laugh?. In *Proceedings of the ACL-IJCNLP 2009 conference short papers.* 161–164.

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[7] Nadia K 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* 52, 1 (2015), 1–4.

[8] Dipto Das and Anthony J Clark. 2019. Satire vs fake news: You can tell by the way they say it. In *2019 First International Conference on Transdisciplinary AI (TransAI)*. IEEE, 22–26.

[9] Janaína Ignácio de Morais, Hugo Queiroz Abonizio, Gabriel Marques Tavares, André Azevedo da Fonseca, and Sylvio Barbon Jr. 2019. Deciding among Fake, Satirical, Objective and Legitimate news: A multi-label classification system. In *Proceedings of the XV Brazilian Symposium on Information Systems*. 1–8.

[10] Sohan De Sarkar, Fan Yang, and Arjun Mukherjee. 2018. Attending sentences to detect satirical fake news. In *Proceedings of the 27th international conference on computational linguistics*. 3371–3380.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] D Funke. 2017. A satirical fake news site apologized for making a story too real. *Poynter. Retrieved from https://www. poynter. org/news/satirical-fake-news-site-apologized-making-story-too-real* (2017).

[13] Johannes Fürnkranz. 1998. A study using n-gram features for text categorization. *Austrian Research Institute for Artifical Intelligence* 3, 1998 (1998), 1–10.

[14] Jennifer Golbeck, Matthew Mauriello, Brooke Auxier, Keval H Bhanushali, Christopher Bonk, Mohamed Amine Bouzaghrane, Cody Buntain, Riya Chanduka, Paul Cheakalos, Jennine B Everett, et al. 2018. Fake news vs satire: A dataset and analysis. In *Proceedings of the 10th ACM Conference on Web Science*. 17–21.

[15] Tarek Hamdi, Hamda Slimi, Ibrahim Bounhas, and Yahya Slimani. 2020. A hybrid approach for fake news detection in twitter based on user features and graph embedding. In *International conference on distributed computing and internet technology*. Springer, 266–280.

[16] Benjamin D Horne and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Eleventh international AAAI conference on web and social media*.

[17] Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences* 9, 19 (2019), 4062.

[18] Junaed Younus Khan, Md Tawkat Islam Khondaker, Sadia Afroz, Gias Uddin, and Anindya Iqbal. 2021. A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications* 4 (2021), 100032.

[19] Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. Mvae: Multimodal variational autoencoder for fake news detection. In *The world wide web conference*. 2915–2921.

[20] DO Klein and JR Wueller. 2017. Fake news: A legal perspective. SSRN.

[21] Sebastian Kula, Michał Choraś, and Rafał Kozik. 2019. Application of the BERT-based architecture in fake news detection. In *Computational Intelligence in Security for Information Systems Conference*. Springer, 239–249.

[22] David Leonhardt and Stuart A Thompson. 2017. Trump's lies. *New York Times* 21 (2017).

[23] Or Levi, Pedram Hosseini, Mona Diab, and David A Broniatowski. 2019. Identifying nuances in fake news vs. satire: using semantic and linguistic cues. *arXiv preprint arXiv:1910.01160* (2019).

[24] Smithsonian Magazine. 2011. The science of sarcasm? yeah, right. https://www.smithsonianmag.com/science-nature/the-science-of-sarcasm-yeah-right-25038/

[25] Shafayat Bin Shabbir Mugdha, Sayeda Muntaha Ferdous, and Ahmed Fahmin. 2020. Evaluating machine learning algorithms for bengali fake news detection. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 1–6.

[26] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*. 2931–2937.

[27] Victoria L Rubin, Yimin Chen, and Nadia K Conroy. 2015. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology* 52, 1 (2015), 1–4.

[28] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[29] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter* 19, 1 (2017), 22–36.

[30] Matt Swayne. 2012. Satire is shaping the next generation of American citizens. https://www.psu.edu/news/research/story/satire-shaping-next-generation-american-citizens/

[31] William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648* (2017).