IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2320-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

|DOI:10.15680/IJIRSET.2021.1007017|

Classifying Fake News Articles Using Natural Language Processing and Machine Learning

Dr. G. Arun Sampaul Thomas [1], T.N.Nithish Kumar [2], K.Paramesh [3], K.V.Gopi [4], P.Niranjan [5]

Associate Professor, Department of Computer Science Engineering, J B Institute of Engineering and Technology, Moinabad, Hyderabad, Telangana, India [1]

B.Tech Student, Department of Computer Science Engineering, J B Institute of Engineering and Technology, Moinabad, Hyderabad, Telangana, India [2,3,4,5]

ABSTRACT: In this current generation where the news can be spread very quickly because of the internet, the impact of fake news has also been increasing. Because of social media, these fake news are distributed on all the social media sites and finds their way too on to the television and other platforms. The spreading of fake news is directing people in the wrong direction and it is creating a lot of disturbance in society. It has been affecting the results of the election and also affecting the law and order. To overcome this issue, in this paper, we analysed different existing fake news classifying models and came up with a solution where we use Natural Language Processing techniques such as Count Vectorization or Tfid Vectorization and Machine Learning algorithms such as Logistic Regression and Multinomial Naïve Bayes for classification of fake news with high accuracy. This paper contains six modules. Collection of data, Preprocessing, Exploratory data analysis, Natural Language processing, Training, and Testing.

KEYWORDS: Machine learning, Natural language processing, Logistic Regression, Multinomial Naïve Bayes, Count Vectorization, Tfidf Vectorization.

I. INTRODUCTION

Fake news is misleading information presented as news. It is resulting in damaging the reputation of a person or community. However, if we consider recent incidents that occurred due to the spread of fake news on COVID-19, it led to many problems. Some of them led to the collapse of different businesses. Many individuals and organizations are spreading fake news unknowingly. Before realizing whether the news is fake or not, it is creating a lot of havoc in society. Fake news can reduce the impact of real news. Multiple strategies are being developed to prevent fake news but they are not that effective.

We are classifying fake news articles using Natural language processing and Machine learning. In this first, we collect data from various sources like the internet, social media, etc. We create datasets from the collected data. Those datasets are helpful to detect fake news. In this project, we use Natural Language processing that focuses on the interaction between human language and computer language. Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation. With the help of machine learning, we perform training and testing on the datasets using different algorithms like logistic regression and multinomial naive bayes and find the accuracy of the model. If the accuracy is much as we expected then we allow articles to pass through the build model to classify fake news present in it.

II. RELATED WORK

Uma Sharma, Shankar M. Patil, and Sidarth Saran introduced a fake news classification system using Machine learning techniques where they used N-grams technique for vectorizing the data [1]. They didn't convert the text which is in the upper case into lower case. It affected the preprocessing. They used Random Forest and Naïve Bayes algorithms but the accuracy that model achieved is below 80%. Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat, and Justin Hsu researched on the vulnerability of existing fake news models and how data can be manipulated to mislead the fake news classifying tools [2].

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

|DOI:10.15680/IJIRSET.2021.1007017|

In this paper, we directly scrape data from the internet i.e. the news publishing sites so that the data can't be manipulated. In the past, for example, in the 2016 United states elections, fake news had created a large impact on people's perception and it has resulted in affecting the election result. Research also suggests that combating misinformation is a difficult challenge (for reviews, see, e.g., Flynn, Nyhan, and Reifler 2017; Lewandowsky et al. 2012). In recent COVID times, we can see how these fake news is affecting the lives of people. In 2020, there was news that hens are also spreading COVID. It is not scientifically proven but this news has created havoc in public. Everyone stopped eating chicken and the chicken prices had fallen drastically. It created a huge loss in the poultry business. In 2020, we had another fake news regarding COVID that if the temperature is greater than 25 degrees Celsius, then the virus will not spread and it'll die. This news made people more complacent and many people didn't understand the seriousness of the virus and eventually paid the price for it. Like this, these fake news are affecting people's lives in many different ways. To overcome this issue, we made this model, which is more efficient than many other existing fake news classification systems.

III. METHODOLOGY

The Methodology consists of six stages. They are Collection of Datasets, Preprocessing, Exploratory data analysis, Natural Language Processing, Training, and Testing.

Step 1 – Collection of data: In this paper, both real news as well as fake news are collected and a .csv file is created with the collected data. The real news and fake news are stored in different files so that the operations can be performed easily on them. Here, we will scrape the data from news publishing sites using PushshiftAPI() function, so that we can get the latest news in our project. This scraped data is converted into a .csv file so that it can be read easily by the machine.

Step 2 – Preprocessing: This data has lots of redundant values and it also contains null values. So by using pandas and Regular Expressions, all the redundant values and also the null values are removed. We also remove the extra spaces and the entire text is converted into lower case text so that the data can be analyzed properly and the efficiency of the product also increases. While pre-processing the data, special characters are also removed so that the data can be analysed to the core.

Step 3 – Exploratory Data Analysis: After cleaning the data, Exploratory Data Analysis (EDA) is performed so that the data can be understood easily. Libraries like Matplotlib and Seaborn is used to visualize the data from various dimensions. This helps us understand the data completely and also understand the data from various dimensions. For example, if the Author and Number of fake news entities are compared, we'll get an idea about which author publishes more fake articles and which author publishes fewer fake articles. Similarly, if the Domain and Number of Fake news entities are compared, we can understand which domain publishes more fake news.

Step 4 – Applying NLP techniques: As the next step, NLP i.e. Natural Language Processing operations are performed on the datasets. We concatenate both the datasets and we map all the real news and fake news with 0 and 1 values respectively. Now the dataset is split into two datasets so that one dataset can be used to create the model by training an algorithm and the other dataset can be used to test the model. In this paper, data is split in the ratio of 7:3 i.e. 70% of the data can be used to train the model and the remaining 30% is used to test the model. NLP algorithms such as Count Vectorization or Tfidf Vectorization are applied to this data. It helps us in getting the association between the words and their result i.e. whether it is fake or not. Using Count Vectorization we convert the data which is in the form of words into numerical i.e. we find the frequency of each word in that document so that we can give these converted datasets to the predictive algorithms for training and testing. Similarly, Tfidf vectorization also finds the frequency of the words but in the tfidf vectorization, we can set boundaries or limits to the frequency of the words so that we can minimize the overload on the model.

Step 5 – Training and Testing: As the next step, Logistic Regression and Multinomial Naïve Bayes predictive algorithms are used for training and testing. In Logistic regression, the sigmoid function is used i.e. $Y = 1 / 1+e^{-z}$ for prediction whereas in Multinomial Naïve Bayes, the posterior probability is used for prediction.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

|DOI:10.15680/IJIRSET.2021.1007017|

$$Posterior\ Probability = \frac{Conditional\ Probability * Prior\ Probability}{Predictor\ Prior\ Probability}$$

This equation is used in Multinomial Naïve Bayes for prediction.

IV. EXPERIMENTAL RESULTS

In figure 1 where the Count Vectorization technique and Logistic Regression algorithm are used, the accuracy of 83.4% is achieved, whereas, in figure 2 where Tfid Vectorization and Logistic Regression is used, the accuracy of 82.6% is achieved.

In figure 3, Count Vectorization and Multinomial Naïve Bayes techniques are used and we got the highest accuracy i.e. 85.8% which is greater than the existing models. Count Vectorization finds the frequency of the words in the document and then it is given as input to the Machine learning algorithm for prediction. Multinomial Naïve Bayes is a part of Naïve Bayes algorithm. It is superior to Naïve Bayes. It analyzes by taking multinomial data and predicts accordingly. This algorithm is mostly used for language processing as it gives efficient and accurate results compared to other algorithms. So, this is the reason we got the highest accuracy for this model.

This model also handles a large amount of data and it doesn't crash even it is overloaded. In this paper, we trained and tested this model with more than 15000 examples so that it can handle the overload.

The below figures show the accuracy of different models which are built using different algorithms.

Model 1: CountVectorizer & Logistic Regression (Best Coefficient Interpretability)

```
8]: pipe = Pipeline([('cvec', CountVectorizer()),
                     ('lr', LogisticRegression(solver='liblinear'))])

pipe_params = {'cvec__stop_words': [None, 'english', custom],
               'cvec__ngram_range': [(1,1), (2,2), (1,3)],
               'lr__C': [0.01, 1]}

grid_search = GridSearchCV(pipe, param_grid=pipe_params, cv=3)
grid_search.fit(X_train, y_train);
print("Best score:", grid_search.best_score_)
print("Train score", grid_search.score(X_train, y_train))
print("Test score", grid_search.score(X_test, y_test))

grid_search.best_params_

Best score: 0.8290452572845629
Train score 0.9578941930150857
Test score 0.8341599504029759

8]: {'cvec__ngram_range': (1, 1), 'cvec__stop_words': None, 'lr__C': 1}
```

Fig 1: Model built using Count Vectorization and Logistic Regression

Model 2: TfidfVectorize & Logistic Regression

```
[49]: pipe = Pipeline([('tvect', TfidfVectorizer()),
                       ('lr', LogisticRegression(solver='liblinear'))])


pipe_params = {'tvect__max_df': [.75, .98, 1.0],
               'tvect__min_df': [2, 3, 5],
               'tvect__ngram_range': [(1,1), (1,2), (1,3)],
               'lr__C': [1]}

gs = GridSearchCV(pipe, param_grid=pipe_params, cv=3)
gs.fit(X_train, y_train);
print("Best score:", gs.best_score_)
print("Train score", gs.score(X_train, y_train))
print("Test score", gs.score(X_test, y_test))

gs.best_params_

Best score: 0.8251704897706138
Train score 0.9070055796652201
Test score 0.8268753874767514
```

Fig 2: Model built using Tfid Vectorization and Logistic Regression

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 7, July 2021 ||

|DOI:10.15680/IJIRSET.2021.1007017|

Model 3: CountVectorizer & MultinomialNB (Best Accuracy Score)

```
[50]: pipe = Pipeline([('cvec', CountVectorizer()),
                       ('nb', MultinomialNB())])


pipe_params = {'cvec__ngram_range': [(1,1),(1,3)],
               'nb__alpha': [.36, .6]}

gs = GridSearchCV(pipe, param_grid=pipe_params, cv=3)
gs.fit(X_train, y_train);
print("Best score:", gs.best_score_)
print("Train score", gs.score(X_train, y_train))
print("Test score", gs.score(X_test, y_test))

gs.best_params_
```

```
Best score: 0.8585968175242819
Train score 0.9955569332506716
Test score 0.8685678859268444
```

Fig 3: Model built using Count Vectorization and Multinomial Naïve Bayes

V. CONCLUSION

In this paper, the classifying fake news technique (which is built using NLP and ML) is implemented on articles that we collected randomly. Our Model successfully classified the fake news from the articles with an accuracy of 85.8% which is better than other existing models. We have applied our model on more than 15000 articles and found that it successfully detected the text region where the fake news has been used.

REFERENCES

1. Uma Sharma, Shankar M. Patil, Sidarth Saran, “Fake news detection using Machine Learning Algorithms”, IJCRT ISSN 2320-2882, 06-06-2020.
2. Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat and Justin Hsu, “Fake News Detection via NLP is Vulnerable to Adversarial Attacks”, WISC, 01-06-2019.
3. Granik, M. and Mesyura, V. (2017). Fake news detection using naive Bayes classifier. In IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON).
4. Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. (2017). Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter.
5. Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). Automatic deception detection: methods for finding fake news.
6. S. Feng, R. Banerjee, and Y. Choi, “Syntactic stylometry for deception detection,” in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Association for Computational Linguistics, 2012, pp. 171–175.
7. Shlok Gilda, Department of Computer Engineering, Evaluating Machine Learning Algorithms for Fake News Detection,2017 IEEE

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details