IJIEMR Transactions, online available on 29th Aug 2021.

Link :http://www.ijiemr.com/downloads.php?vol=Volume-10&issue=ISSUE-08

DOI: **10.48047/IJIEMR/V10/I08/19-13**

Title:- **FAKE NEWS DETECTION USING MACHINE LEARNING NLP**

Volume 10, Issue 08, Pages:207-214

Paper Authors
Mr.Polasi.Sudhakar[1], Sanapala Neeraja [2], Badipatla Snigdha [3] , Pilli Ashok [4] , Kondapalli Keerthana [5]

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.com

# FAKE NEWS DETECTION USING MACHINE LEARNING NLP

Mr.Polasi.Sudhakar[1], Sanapala Neeraja[2], Badipatla Snigdha[3], Pilli Ashok[4], Kondapalli Keerthana[5]
[1]Assistant Professor, Dept. of CSE, [2]17ME1A05B1, [3]17ME1A0564, [4]17ME1A05A7, [5]17ME1A0591
Ramachandra College of Engineering, A.P., India

## ABSTRACT

Fake news is one of the biggest issues in the current trend of the internet and social media. Fake news is misleading people by spreading news from the non-reputable sources. To stop spread of the fake news to some extent a (Natural Language Processing) NLP and Machine Learning based web application is used for detecting fake news. Detecting the fake news is a classic text classification problem. The data is collected from the Kaggle data sets, after that pre-processing is done. And, for the classification purpose Multinomial Naive Bayes classifier is used and TF-IDF Vectorizer is used from which a model is built that can differentiate "Real" news and "Fake" news based on training set and test set. And accuracy from the test data set is almost 95.71%. Technologies used are HTML, Bootstrap, Django, and some of the machine Learning libraries from Python.

## Introduction

As the Internet continues to grow in size and importance, the quantity and impact of online reviews is increasing continuously. Reviews can influence people across a wide range of industries, but they are particularly important in e-commerce, where comments and reviews on products and services are often the most convenient, if not the only, way for a buyer to decide whether to buy them. Online reviews can be generated for a variety of reasons. Online retailers and service providers may often ask their customers to provide feedback on their experience with the products or services they have purchased in order to improve and enhance their businesses. Customers may also feel inclined to review a product or service if they had an exceptionally good or bad experience with it. While online reviews can be helpful, blind trust of these reviews is dangerous for both the seller and buyer. Many looks at online reviews before placing any online order; however, the reviews may be poisoned or faked for profit or gain, thus any decision based on online reviews must be made cautiously. Furthermore, business owners might give incentives to whoever writes good reviews about their merchandise or might pay Someone write bad reviews of the products or services of their competitor. These fake reviews are considered spam review and due to the importance of reviews can have a great impact on the online marketplace. Someone to write bad reviews about their competitor's products or services. As review spam is a pervasive and harmful issue, it is an important but challenging issue to develop methods to help businesses and consumers distinguish true reviews from fake ones. It's not only the case related to e-commerce online reviews there is fake news that is leading people to wrong paths and wrong perceptions because of which people may loose trust in internet in upcoming years.

## Existing System

Initially context related n-grams and shallow parts-of-speech tagging has been used for classification tasks. Later, Deep Syntax analysis using Probabilistic Context Free Grammars (PCFG) have used in combination with n-gram methods. An N-gram is a sequence of words. n-grams are used for a variety of things. Some examples include auto completion of sentences (such as the one we see in Gmail these days), auto spell check (yes, we can do that as well), and to a certain extent, we can check for grammar in a given sentence. Suppose during the sentence completion Suppose we're calculating the probability of word "w1" occurring after the word "w2," then the formula for this is as follows:

count (w2 w1) / count(w2)

which is the number of times the words occurs in the required sequence, divided by the number of the times the word before the expected word occurs in the corpus. Using these n-grams and the probabilities of the occurrences of certain words in certain sequences could improve the predictions of auto completion systems. Similarly, we use can NLP and n-grams to train voice-based personal assistant bots. Deep syntax can be analysed using Probabilistic context-free grammar (PCFG). Syntax structures are described by changing sentences into parse trees. Nouns, verbs etc. are rewritten into their syntactic constituent parts. Probabilities are assigned to the parse tree.

Limitations associated with existing System

1. Machine learning methods for deception detection are mainly focused on classifying online reviews and publicly available social media posts.

2. This process may not yield exact accurate results.

3. It is not good for large data sets and it may even fail.

## Proposed System

In this, a model is built based on count vectorizer or TFIFD vectorizer. Since this problem is a kind of text classification, a Multinomial Naive Bayes classifier is used for text-based processing. To check the accuracy of the model confusion matrix or accuracy score is used.

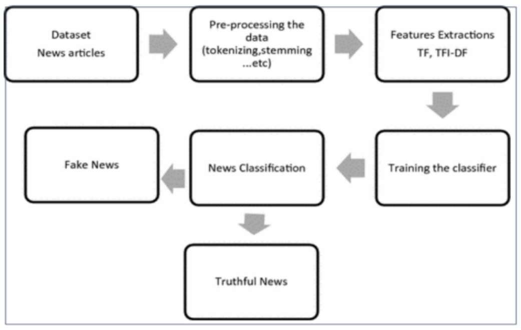
Advantages

1. The system mainly focuses on Pre-processing of Data.

2. The classification process is easier as compared to existing model.

3. The accuracy obtained in this model is nearly 95.1%.

System Architecture



Datasets Splitting the data is the most essential step in machine learning. We train our model on the trainset and test our data on the testing set. We split our data in train and test using the train_test_split function from Scikit learn. We split our 80% data for the training set and the remaining 20% data for the testing set. Test Dataset A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal over fitting has taken place. A better fitting of the training dataset as opposed to the test dataset usually points to over fitting. A test set is therefore a set of examples used only to assess the performance (i.e., generalization) of a fully specified classifier. Validation Dataset A validation dataset is a set of examples used to tune the hyper parameters (i.e., the architecture) of a classifier. It is sometimes also called the development set or the "dev set". In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation dataset in addition to the training and test datasets. Training Dataset

Pre-processing the data Cleaning Data We can't use text data directly because it has some unusable words and special symbols and many more things. If we used it directly without cleaning then it is very hard for the ML algorithm to detect patterns in that text and sometimes it will also generate an error. So that we have to always first clean text data. In this project, we are making one function 'cleaning data' which cleans the data. Lemmatization Stemming and Lemmatization are the techniques to reduce the words to their stems or roots. The main advantage of this step is to reduce the size of the vocabulary. Convert the word or token in its Base form.

Feature extraction TF-IDF Vectorizer (Term Frequency * Inverse Document Frequency) 1.Term Frequency: The number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse Document Frequency: The log of the number of documents divided by the number of documents that contain the word w. Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = log(\frac{N}{df_t})$$

Finally, TF-IDF vectorizer

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Naive Bayes The Naive Bayes Classifier technique is based on the Bayesian theorem and is particularly suited when then high dimensional data

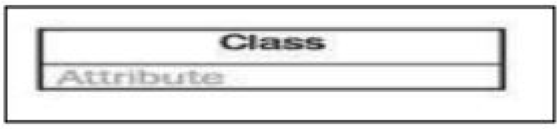
$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

| | |
|---|---|
| $A, B$ | = events |
| $P(A\|B)$ | = probability of A given B is true |
| $P(B\|A)$ | = probability of B given A is true |
| $P(A), P(B)$ | = the independent probabilities of A and B |

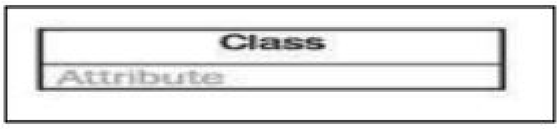
Multinomial Naive Bayes It is used for classification when is in discrete form. It is very useful in text processing. Each text will be converted to a vector of word count. It cannot deal with negative numbers. It is predefined in Scikit Learn Library. So, we can import that class in our project then we create an object of Multinomial Naive Bayes Class. 1. Fit the classifier on our vectorized train data 2. When the classifier fitted successfully on the training set then we can use the predict method to predict the result on the test set

Classification Metrics: To check how well our model we use some metrics to find the accuracy of our model. There are many types of classification metrics available in Scikit learn 1. Confusion Matrix 2. Accuracy Score 3. Precision 4. Recall 5. F1-Score

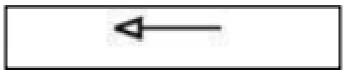
Notations, Class A class represents a relevant concept from the domain, a set of persons, objects, or ideas that are depicted in the IT system.



Attribute an attribute of a class represents a characteristic of a class that is of interest for the user of the IT system.

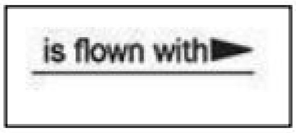


Generalization is a relationship between two classes: a general class and a special class.

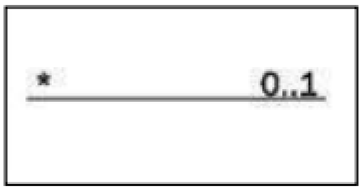


Refer to Generalization, Specialization, Inheritance

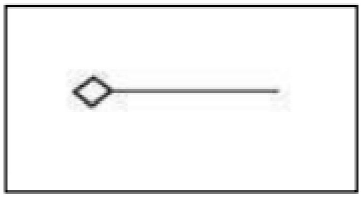
Association An association represents a relationship between two classes. An Association represents a family of links. A binary association (with two ends) is normally represented as a line. An association can link any number of classes. An association with three links is called a ternary association. Different types of association are:  One-To-One Association  One-To-Many Association  Many-To-Many Association



Multiplicity A multiplicity allows for statements about the number of objects that are involved in an association.



Aggregation An aggregation is a special case of an association (see above) meaning "consists of". The diamond documents this meaning; a caption is unnecessary. Each end of an associate can be labelled by a set of integers indicating number of links that legitimately originate from the instance if the class connects to association end. This set of integers is called multiplicity of associate end.

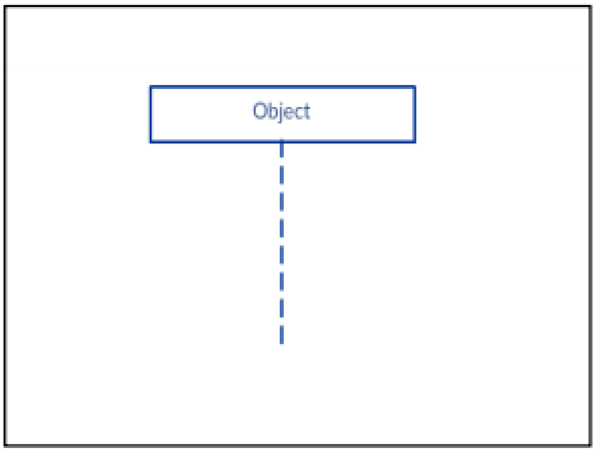


Use Case Diagram In the Dynamic en-route filtering scheme the dataflow diagram describes the En-Route node receives a report from the source node or the lower associated en-route node and check the integrity of the received report by means of the MAC enclosed in the report. If the Verification succeeds, then forward the report otherwise drop he report.

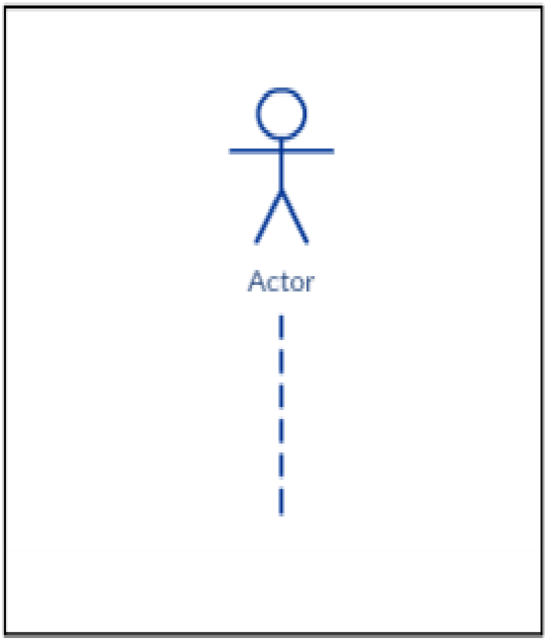
Sequence Diagram

The Sequence diagram shows the order in which interactions take place. The diagram is used to depict the interaction between several objects in a system. Sequence Diagrams are used mostly by software developers to note down and understand the requirements of new and pre-existing systems. Many Businessmen also use these diagrams to understand and establish systems too.

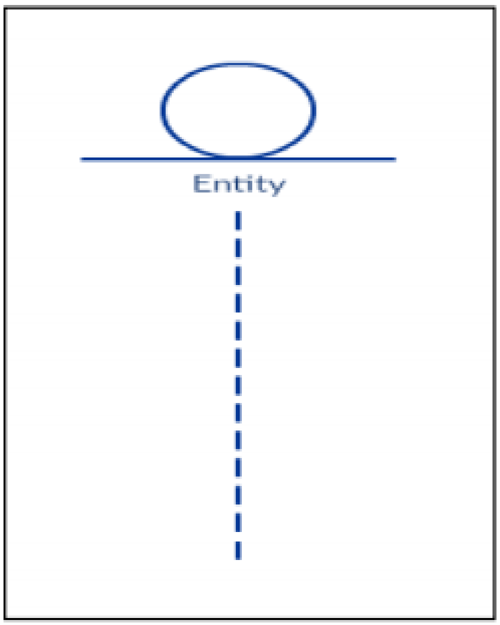
A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.



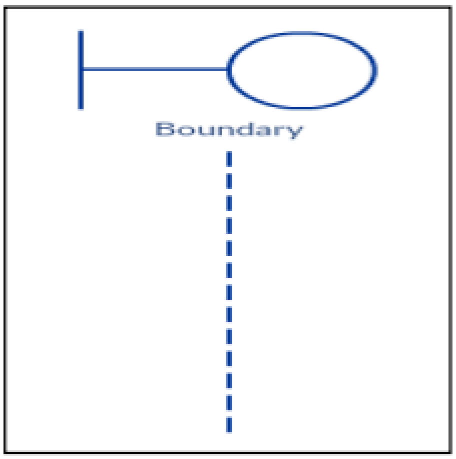
A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the diagram. No two lifeline notations should overlap each other. They represent the different objects or parts that interact with each other in the system during the sequence. A lifeline notation with an actor element symbol is used when the particular sequence diagram is owned by a use case.



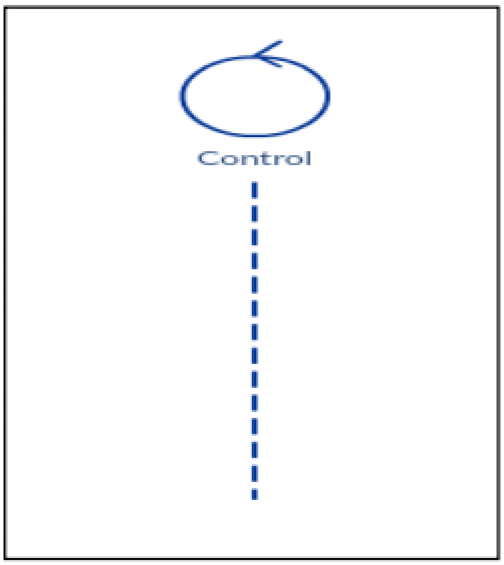
A lifeline with an entity element represents system data. For example, in a customer service application, the Customer entity would manage all data related to a customer.



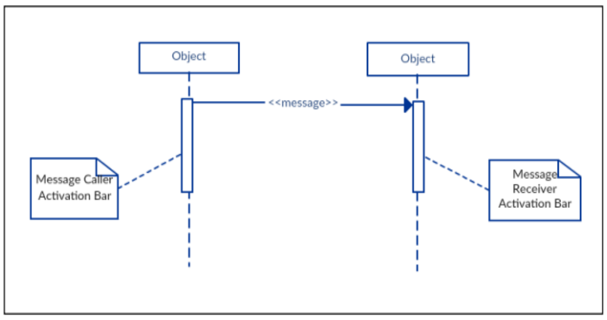
A lifeline with a boundary element indicates a system boundary/ software element in a system; for example, user interface screens, database gateways or menus that users interact with, are boundaries.



And a lifeline with a control element indicates a controlling entity or manager. It organizes and schedules the interactions between the boundaries and entities and serves as the mediator between them.



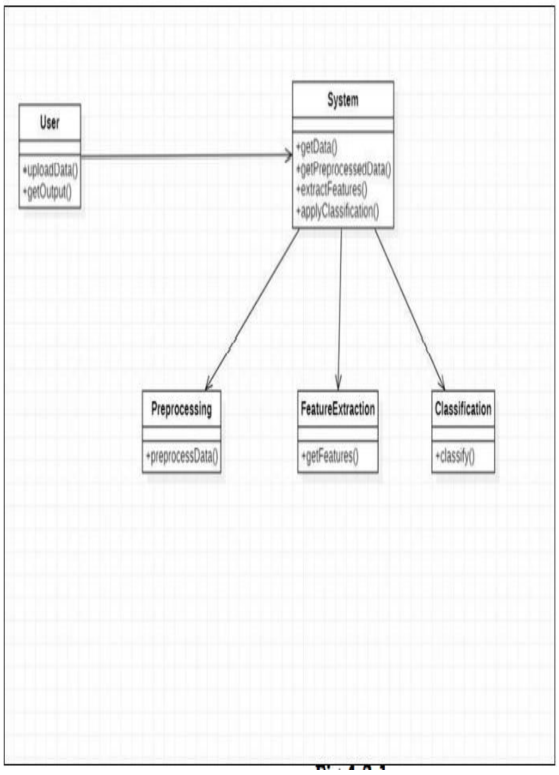
Activation bar is the box placed on the lifeline. It is used to indicate that an object is active (or instantiated) during an interaction between two objects. The length of the rectangle indicates the duration of the objects staying active. In a sequence diagram, an interaction between two objects occurs when one object sends a message to another. The use of the activation bar on the lifelines of the Message Caller (the object that sends the message) and the Message Receiver (the object that receives the message) indicates that both are active/is instantiated during the exchange of the message.


Object
Object
<<message>>
Message Caller Activation Bar
Message Receiver Activation Bar

Message Arrows

An arrow from the Message Caller to the Message Receiver specifies a message in a sequence diagram. A message can flow in any direction; from left to right, right to left or back to the Message Caller itself. While you can describe the message being sent from one object to the other on the arrow, with different arrowheads you can indicate the type of message being sent or received. The message arrow comes with a description, which is known as a message signature, on it. The format for this message signature is below. All parts except the message name are optional.

Class diagram



## Conclusion

In this project, a model is built that is used to detect fake news and real news. To build the model Multinomial Naïve Bayes Classifier is used. Moreover, to vectorize the text data, we used the TF-IDF vectorizer. By taking the input data from Kaggle data sets we fed that into the model. And predict whether data is real or fake along with accuracy. As we are using Multinomial Naïve Bayes, 95.1% accuracy is obtained. As it is useful for predicting fake news we can stop spreading of fake news to some extent. The main contribution of this project is support for the idea that machine learning could be useful in a novel way for the task of classifying fake news. Our findings show that after much pre-processing of relatively small data set, our model is able to pick up on a diverse set of potentially subtle language patterns that a human may (or may not) be able to detect. Many of these language patterns are intuitively useful in humans manner of classifying fake news. Even if a human could detect these patterns, they are not able to store as much information and therefore, may not understand the complex relationships between the detection of these patterns and the decision for classification. This application could be tool for humans trying to classify the fake news, to get indication of which words might cut them into correct classification. It could be useful in researches trying to develop improved models through the use of improved and enlarged dataset for different parameters, etc., The application also provides to see manually how changes in the body text affect the classification.

**Future Work**Through this work, we have shown that machine learning certainly does have the capacity to pick up on sometimes subtle language pattern s that may be difficult for humans to pick up on. The next steps involved in this project come in two different aspects. The first aspect is that could be improved in this project is augmenting and increasing the size of the data set. We feel that more data would be beneficial in ridding the model of any bias based on specific patterns in the source. There is also question or not the size of our dataset is sufficient. The second aspect in which this project could be expanded is by comparing it to humans performing the same task. Computing the accuracies will be beneficial in deciding whether or not the dataset is representative of how difficult the task of separating fake news from real news. If humans are more accurate than the model, it may mean that we need to choose more deceptive fake news examples. Because we acknowledge that this is only one tool in a toolbox that would really be required for an end-to-end system for classifying fake news, we expect that its accuracy will never reach perfect. However, it may be beneficial as a stand-alone application if its accuracy is higher than human accuracy at the same task. In addition to comparing accuracy to human accuracy, it would also be interesting to compare the phrases that a human would point out if asked when they based their classification decision on. Then, we could quantify how similar those patterns are to those humans find indicative of fake and real news.

**REFERENCES** 1. Arjun Mukherjee, VivekVenkataraman, Bing Liu, and Natalie Glance. What Yelp Fake Review Filter Might Be Doing. Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013), July 8-10, 2013, Boston, USA. 2. B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but toughto-beat baseline for the fake news challenge stance detection task," 2017, https://arxiv.org/abs/1707.03264. 3. Kaggle, Fake News Detection, Kaggle, San Francisco, CA, USA, 2018, https://www.kaggle.com/jruvika/fake-news-detection. 4. M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, H. A. Najada, "Survey of review spam detection using machine learning techniques", Journal of Big Data, vol. 2, no. 23, Dec 2015. 5. V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," 2017, https://arxiv.org/abs/1708.07104.