A

Seminar Report

On

Machine Learning for Fake News Detection

By

Aditya Rajesh Padmawar

Roll No: 305086

Under the Guidance of

Prof. N.A.Mhetre

Sinhgad Institutes

Department of Computer Engineering

Sinhgad College of Engineering, Pune-41

Accredited by NAAC

SAVITRIBAI PHULE PUNE UNIVERSITY
2018-19

Sinhgad Technical Education Society,

Department of Computer Engineering

Sinhgad College of Engineering, Pune - 41

Sinhgad Institutes

Date:

Certificate

This is to certify that Mr. Aditya Rajesh Padmawar has successfully completed his seminar work on "Machine Learning for Fake News Detection " at Sinhgad College of Engineering , Pune in the partial fulfillment of the Graduate Degree course in T.E. at the department of Computer Engineering, in the academic Year 2018-2019 Semester VI as prescribed by the Savitribai Phule Pune University.

Prof. N.A.Mhetre
Guide
.

Prof. M.P.Wankhade
Head of Department
Department of Computer Engineering

Dr. S. D. Lokhande
Principal
Sinhgad College of Engineering

Acknowledgements

I would like to express my deepest appreciation to all those who helped me for smooth conduct of this seminar. I would like to express my gratitude to my seminar guide Prof. N.A.Mhetre who has invested her full effort in guiding me and whose contribution helped me to co-ordinate my seminar on Machine Learning For Fake News Detection. I am grateful to our respected HOD (Comp Dept.) Prof.M.P.Wankhade and Principal(SCOE) Dr. S.D.Lokhande for providing all necessary facilities. Furthermore, I would also like to appreciate the crucial role of our department staff who gave the permission to use all the required equipment. Last but not the least, I would like to thank all the unseen authors of various articles on the internet for updating me about all the latest technological aspects and also for making me aware of ongoing research in my seminar topic.

Aditya Rajesh Padmawar

Roll No.-305086

Abstract

This seminar explores the application of processing techniques for the detection of 'fake news', that is, misleading news stories that come from non-reputable sources. Fake news is a phenomenon which is having a significant impact on our social life, in particular in the political world. There needs to be a solution to eliminate such fake news from the huge world of internet in order to get reliable and unbiased news. This seminar gives a brief idea about the ways in which Fake news can be detected by using machine learning. Fake news detection is an emerging research area which is gaining interest but involves some challenges due to the limited amount of resources (i.e., datasets, published literature) available.

List of Figures

Acronyms

AI	Artificial Intelligence
CFG	Context Free Grammar
DT	Decision Tree
KNN	K-Nearest Neighbour
LSVM	Linear Support Vector Machine
ML	Machine Learning
NBC	Naive Bayes Classifier
NLP	Natural Language Processing
SVM	Support Vector Machine
TF-IDF	Term Frequency-Inverted Document Frequency

Contents

Chapter 1

Introduction

In the recent years, online content has been playing a significant role in swaying users decisions and opinions. Opinions such as online reviews are the main source of information for e-commerce customers to help with gaining insight into the products they are planning to buy. Recently it has become apparent that opinion spam does not only exist in product reviews and customers feedback. In fact, fake news and misleading articles is another form of opinion spam, which has gained traction. Some of the biggest sources of spreading fake news or rumors are social media websites such as Google Plus, Facebook, Twitters, and other social media outlet. Even though the problem of fake news is not a new issue, detecting fake news is believed to be a complex task given that humans tend to believe misleading information and the lack of control of the spread of fake content. Fake news has been getting more attention in the last couple of years, especially since the US election in 2016. It is tough for humans to detect fake news. It can be argued that the only way for a person to manually identify fake news is to have a vast knowledge of the covered topic. Even with the knowledge, it is considerably hard to successfully identify if the information in the article is real or fake. We discuss various techniques to detect fake news, which consists of using text analysis based machine learning classification techniques.

1.1 Motivation

The goal is to attempt to tackle the growing issue of fake news, which has been exacerbated by the wide-spread use of social media. For example, many believe fake news on social media to be a large contributing factor to results of the controversial 2016 US election. We want to create an easy-to-use system to detect the credibility of a users claim or article.

1.2 Timeline/Evolution

Machine Learning technology has been in existence since 1952. It has evolved drastically over the last decade and saw several transition periods in the mid-90s. The data-driven

approach to Machine Learning came into existence during the 1990s. From 1995-2005, there was a lot of focus on natural language, search, and information retrieval. In those days, Machine Learning tools were more straightforward than the tools being used currently. Neural networks, which were popular in the 80s, are a subset of Machine Learning that are computer systems modeled on the human brain and nervous system. Neural networks started making a comeback around 2005. It has become one of the trending technologies of the current decade. According to Gartner's 2016 Hype Cycle for Emerging Technologies, Machine Learning is among the technologies at the peak of inflated expectations and is expected to reach the mainstream adoption in the next 25 years. Technological capabilities such as infrastructure and technical skills also must advance to keep up with the growth of Machine Learning. Machine Learning has been one of the most active and rewarding areas of research due to its widespread use in many areas. It has brought a monumental shift in technology and its applications. Some of the evolution, which made a huge positive impact in real-world problem solving, are highlighted in the following sections.

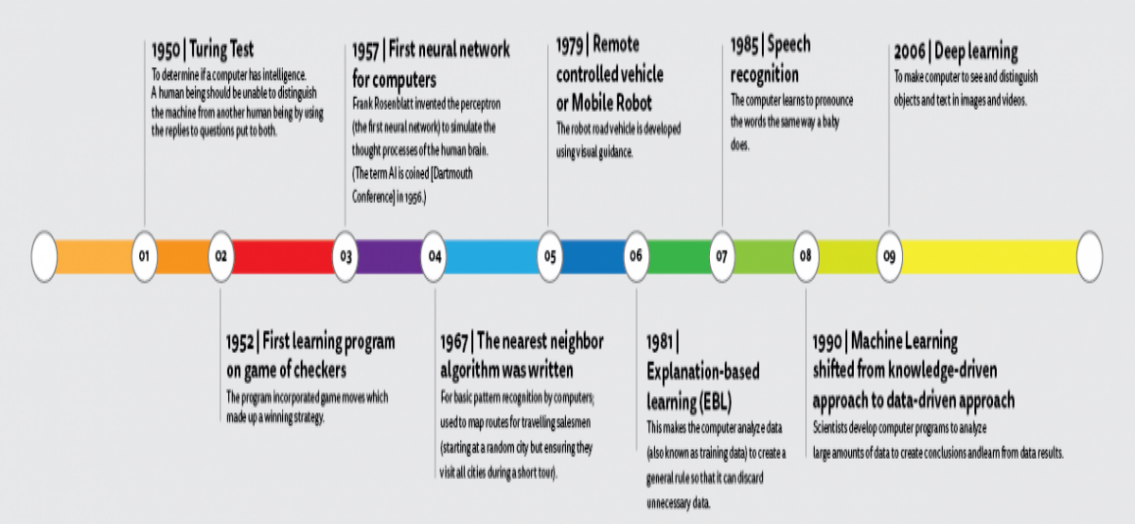

Figure 1.1: Evolution of Learning Machine[2]

Natural language processing (NLP) defines the way or method of connecting computer systems with natural languages such as English. NLP helps computer systems perform the tasks and automate manual processes based on human input, which may be either spoken or written text. For example, shopping by user audio analysis (through speakers like Amazons Alexa) and automating user preference list in a web page based on user

interests. NLP is applied widely to characterize, interpret, or understand the information content of free-form text or unstructured data. It is estimated that 80 percent of the worlds data is unstructured. Hence to handle and get valuable insights from unstructured data, NLP is essential. It allows computer systems to learn and draw insights from data such as email, social media response, audio, and videos, which helps computer systems understand human interaction, human response, and other associated events or activities related to an environment. Unlike the older NLP algorithm generation, which involved manual categorization of text, the modern NLP algorithms are mainly based on statistical Machine Learning. Machine Learning algorithms are used to automatically learn the rules of categorizing the text through the analysis of a corpus (a set of text documents). Many different classes of Machine Learning algorithms such as Decision tree, Support vector machine, Nave Bayes, and so on have been applied to NLP tasks.

1.3 Organization Of Report

The Report is divided into Five Chapters. Chapter 2 covers the Literature Review part of the report. It includes topics viz. Fundamental topics required to present the seminar, Related work done in the same topic and the Conclusion of the survey of the current methods. Chapter 3 includes the main methods which are discussed to properly explain the topic. Chapter 4 covers the Results, i.e. comparison of performance measures. Chapter 5 includes the conclusion of the report.

Chapter 2

Literature Survey

This chapter covers the following: -

2.1 Fundamentals: - It includes the fundamentals of topics needed to present the seminar. Basic concepts are cleared in this topic.

2.2 Related Work: - It includes the various papers and algorithms which have been published until now regarding CRM using various Data mining Activities.

2.3 Survey Conclusion: - This section provides a summary of the pros and cons of some of the papers published until now

2.1 Fundamentals

2.1.1 What is Machine Learning ?

Machine Learning is the branch of computer science that deals with the development of computer programs that teach and grow themselves. According to Arthur Samuel, an American pioneer in computer gaming, Machine Learning is the sub-field of computer science that ”gives the computer ability to learn without being explicitly programmed.” Machine Learning allows developers to build algorithms that automatically improve themselves by finding patterns in the existing data without explicit instructions from a human or developer. Machine Learning relies entirely on the data; the more the data, the more efficient Machine Learning is.

The Machine Learning development approach includes learning from data inputs and evaluating and optimizing the model results. Machine Learning is widely used in data analytics as a method to develop algorithms for making predictions on data. Machine Learning is related to probability, statistics, and linear algebra. Machine Learning is broadly classified into three categories depending on the nature of the learning signal or feedback available to a learning system.

1.Supervised learning:Computer is presented with inputs and their desired outputs. The goal is to learn a general rule to map inputs to the output.

2.Unsupervised learning:Computer is presented with inputs without desired outputs, the

goal is to find structure in inputs.

3.Reinforcement learning:Computer program interacts with a dynamic environment, and it must perform a certain goal without guide or teacher.

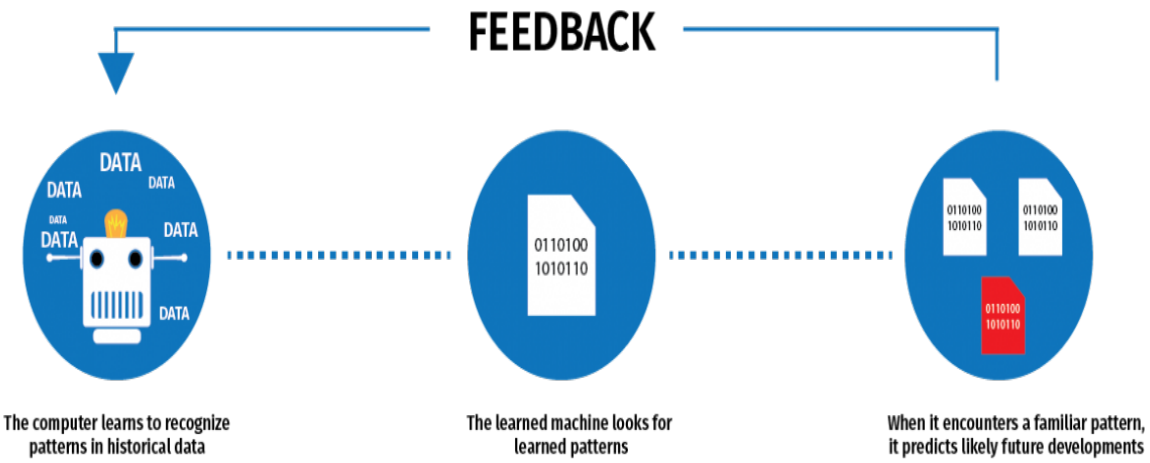

Figure 2.1: Machine Learning Development[6]

2.1.2 How can Machine Learning be used for Detection of Fake News ?

This model is build based on the count vectorizer or a tfidf matrix (i.e) word tallies relatives to how often they are used in other artices in your dataset) can help . Since this problem is a kind of text classification, Implementing a Naive Bayes classifier will be best as this is standard for text-based processing. The actual goal is in developing a model which was the text transformation (count vectorizer vs tfidf vectorizer) and choosing which type of text to use (headlines vs full text). Now the next step is to extract the most optimal features for countvectorizer or tfidf-vectorizer, this is done by using a n-number of the most used words, and/or phrases, lower casing or not, mainly removing the stop words which are common words such as the, when, and there and only using those words that appear at least a given number of times in a given text dataset.

2.2 Related Works

Researchers from MITs Computer Science and Artificial Intelligence Lab (CSAIL) and the Qatar Computing Research Institute (QCRI) believe that the best approach is to focus not only on individual claims, but on the news sources themselves. Using this tack,

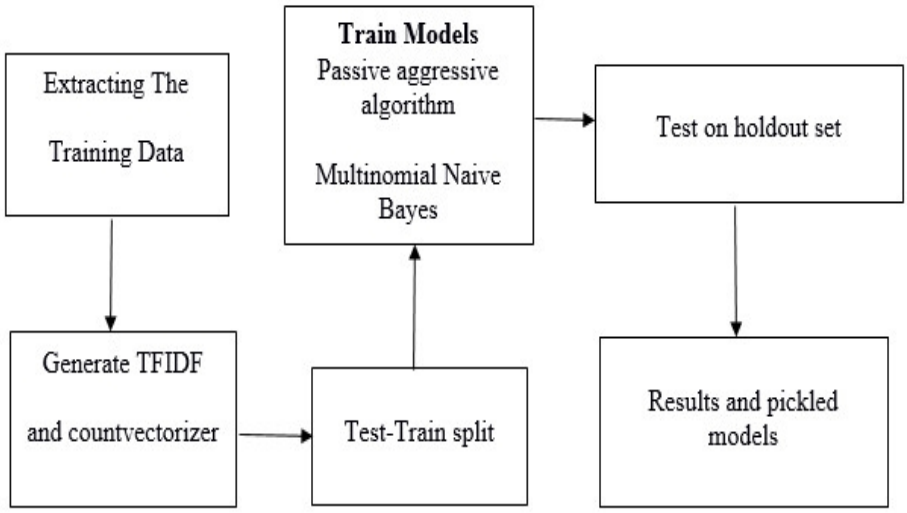

Figure 2.2: fake news[1]

theyve demonstrated a new system that uses machine learning to determine if a source is accurate or politically biased. If a website has published fake news before, theres a good chance theyll do it again, says postdoc Ramy Baly, the lead author on a new paper about the system. By automatically scraping data about these sites, the hope is that our system can help figure out which ones are likely to do it in the first place. Baly says the system needs only about 150 articles to reliably detect if a news source can be trusted meaning that an approach like theirs could be used to help stamp out new fake-news outlets before the stories spread too widely.

2.2.1 IIIT-Delhi students create WhatsFarzi app to detect fake news

WhatsFarzi is the brainchild of three students of IIIT-Delhi. The app will track content on the internet and check it with authentic websites to curb spread of misinformation otherwise known as fake news.

The students of Indraprastha Institute of Information Technology (IIIT), Delhi have developed an app called WhatsFarzi (cab roughly be translated as whatsfake). The app goes through all the internet content and amasses news on a specific topic to verify it from an authentic news portal by using a custom logarithm.[4]

The app is also capable of verifying photos and scanning their authenticity through image tampering algorithms. One of my students started researching on the rapid spread

of fake content on Twitter and Facebook, which inspired him to develop a Google Chrome browser extension for both the platforms.[4]

The continuous research by the team gradually gave birth to WhatsFarzi, which is now helping the vexed Indians to fight back such terrors, said Ponnurangam Kumaraguru, associate professor at IIIT-Delhi.

WhatsFarzi is the brainchild of three students of IIIT-Delhi Madhur Tandon (22), Suryatej Reddy Vyalla (20) and Dhruv Kuchhal (23). All three are studying B.Tech computer science.[4]

2.3 Survey Conclusion

The problem of fake news has gained attention in 2016, especially in the aftermath of the last US presidential elections. Recent statistics and research show that 62 percent of US adults get news on social media. Most of the popular fake news stories were more widely shared on Facebook than the most popular mainstream news stories [14]. A sizable number of people who read fake news stories have reported that they believe them more than news from mainstream media. Dewey claimed that fake news played a huge role in the 2016 US election and that they continue to affect people opinions and decisions.

2.3.1 Advantages

The research showed, that even quite simple Machine Learning algorithm (such as naive Bayes classifier) may show a good result on such an important problem as fake news classification.

Chapter 3

Methodology

3.1 Naive Bayes Classifier and its Usage

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Naive Bayes classifiers are a popular statistical technique of e-mail filtering. They emerged in the middle of the 90s and were one of the first attempts to tackle spam filtering problem.

Naive Bayes typically use bag of words features to identify spam e-mail, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other constructions, syntactic or not), with spam and non-spam e-mails and then using Bayes theorem to calculate a probability that an email is or is not a spam message.

3.2 Mathematical Model of Naive Bayes Classifier for Fake News Detection

The main idea is to treat each word of the news article independently.

As were already mentioned, fake news articles often use the same set of words, which may indicate, that the specific article is indeed a fake news article. Of course, it is impossible to claim that the article is a fake news just because of the fact, that some words appear in it, but these words affect the probability of this fact.

The formula for calculating the conditional probability of the fact, that news article is fake given that it contains some specific word looks as following:

Pr(F—W) = Pr(W—F)Pr(F)/(Pr(W—F)Pr(F)+Pr(W—T)Pr(T)), (1)

where:

Pr(F—W) conditional probability, that a news article is fake given that word W appears in it;

Pr(W—F) conditional probability of finding word W in fake news articles;

Pr(F) overall probability that given news article is fake news article;

Pr(W—T) conditional probability of finding word W in true news articles;

Pr(T) overall probability that given news article is true news article.

This formula is derived from Bayes theorem.

Consider that probabilities Pr(F—W) are known for each word of the news article. Next step is combining this probabilities to get the probability of the fact, that given news article is fake. The formula for this looks as following:

p1 = Pr(F—W1)...Pr(F—Wn), (2)

p2 = (1 Pr(F—W1))...(1 Pr(F—Wn)), (3)

p = p1/(p1 + p2), (4)

where:

n total number of words in the news article;

p1 product of the probabilities that a news article is fake given that it contains a specific word for all of the words in the news article;

p2 same as p1, but complement probabilities are used instead;

Pr(F—W1), Pr(F—W2) Pr(F—Wn) conditional probabilities that a news article is a fake given that words W1, W2, Wn respectively appear in it;

p the overall probability of the fact that given news article is fake.

This formula is often used for spam filtering.

3.3 Implementation Details

3.3.1 Data Pre-processing

Before representing the data using naive bayes model, the data need to be subjected to certain refinements like stop-word removal, tokenization, a lower casing, sentence segmentation, and punctuation removal. This will help us reduce the size of actual data by removing the irrelevant information that exists in the data.

We created a generic processing function to remove punctuation and non-letter characters for each document; then we lowered the letter case in the document. In addition, an n-gram word based tokenizer was created to slice the text based on the length of n.

Stop Word Removal

Stop words are insignificant words in a language that will create noise when used as features in text classification. These are words commonly used a lot in sentences to help connect thought or to assist in the sentence structure. Articles, prepositions and conjunctions and some pronouns are considered stop words. We removed common words such as, a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, the, these, this, too, was, what, when, where, who, will, etc. Those words were removed from each document, and the processed documents were stored and passed on to the next step.

Stemming

After tokenizing the data, the next step is to transform the tokens into a standard form. Stemming simply is changing the words into their original form, and decreasing the number of word types or classes in the data. For example, the words Running, Ran and Runner will be reduced to the word run. We use stemming to make classification faster and efficient. Furthermore, we use Porter stemmer, which is the most commonly used stemming algorithms due to its accuracy.

3.3.2 Features Extraction

One of the challenges of text categorization is learning from high dimensional data. There is a large number of terms, words, and phrases in documents that lead to a high computational burden for the learning process. Furthermore, irrelevant and redundant features can hurt the accuracy and performance of the classifiers. Thus, it is best to perform feature reduction to reduce the text feature size and avoid large feature space dimension. We studied in this research two different features selection methods, namely, Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF). These methods are described in the following.

Term Frequency (TF)

Term Frequency is an approach that utilizes the counts of words appearing in the documents to figure out the similarity between documents. Each document is repre-sented by an equal length vector that contains the words counts. Next, each vector is normalized in a way that the sum of its elements will add to one. Each word count is then converted

into the probability of such word existing in the documents. For example, if a word is in a certain document it will be represented as one, and if it is not in the document, it will be set to zero. Thus, each document is represented by groups of words.

TF-IDF

The Term Frequency-Inverted Document Frequency (TF-IDF) is a weighting metric often used in information retrieval and natural language processing. It is a statistical metric used to measure how important a term is to a document in a dataset. A term importance increases with the number of times a word appears in the document, however, this is counteracted by the frequency of the word in the corpus.

One of the main characteristics of IDF is it weights down the term frequency while scaling up the rare ones. For example, words such as the and then often appear in the text, and if we only use TF, terms such as these will dominate the frequency count. However, using IDF scales down the impact of these terms.

3.3.3 Classification Process

Figure 1 is a diagrammatic representation of the classification process. It starts with preprocessing the data set, by removing unnecessary characters and words from the data. N-gram features are extracted, and a features matrix is formed representing the documents involved. The last step in the classification process is to train the classifier.

We investigated different classifiers to predict the class of the documents.

We investigated specifically six different machine learning algorithms, namely, Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), Linear Support Vector Machines (LSVM), K-Nearest Neighbour (KNN) and Decision Trees (DT). We used implementations of these classifiers from the Python Natural Language Toolkit (NLTK).

We split the dataset into training and testing sets. For instance, in the experiments presented subsequently, we use 5-fold cross validation, so in each validation around 80 percent of the dataset is used for training and 20 percent for testing.

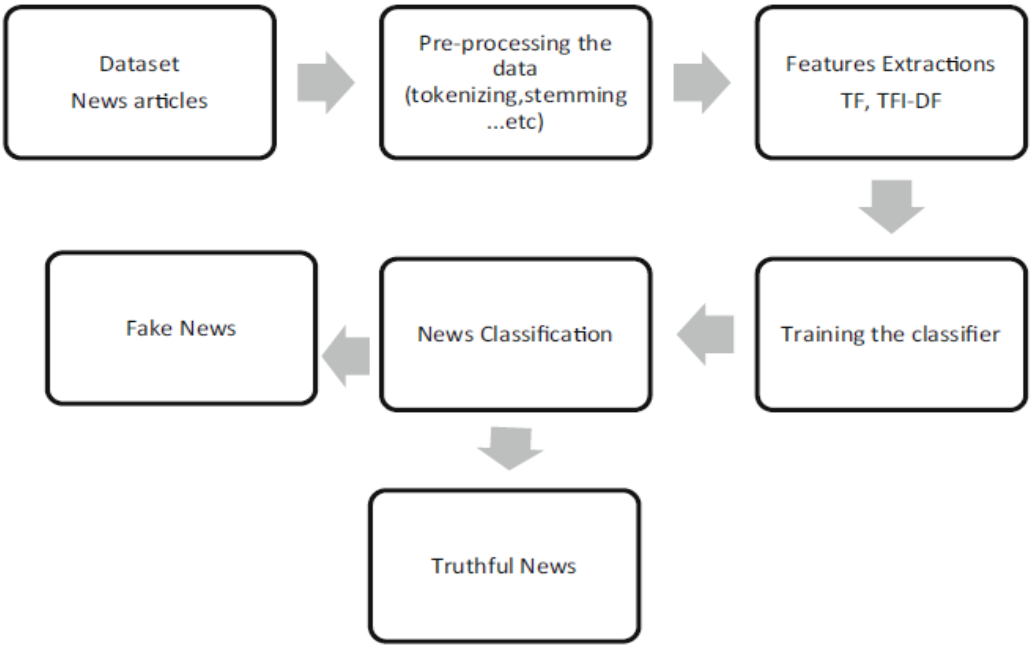

Figure 3.1: Classification Process[1]

Chapter 4

Results

4.1 Experimental Results Analysis

The classification accuracy for true news articles and false news articles is roughly the same, but classification accuracy for fake news is slightly worse. This may be caused by the skewness of the dataset: only 4.9 percent of it are fake news.

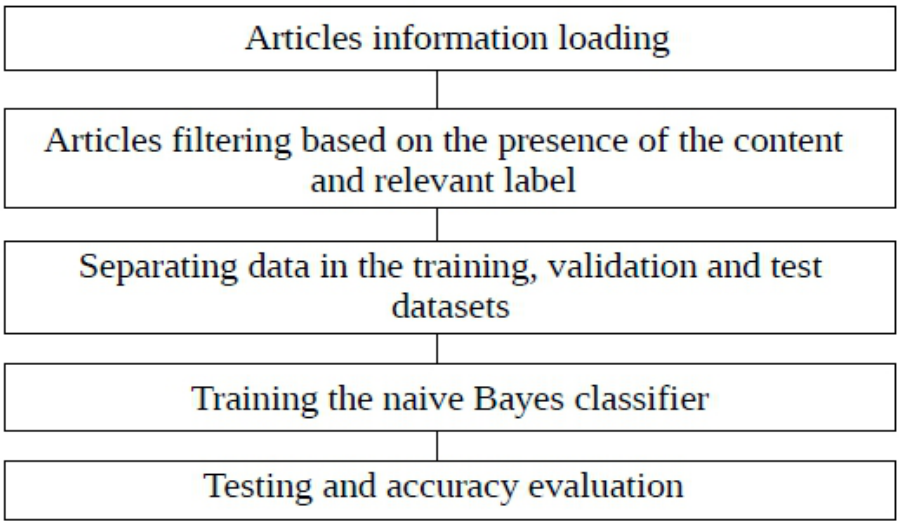

Figure 4.1: Result Flowchart [3]

Lets consider the result as positive, when the classifier classifies the news article as fake. Then:

The number of true positive examples is the number of news articles, correctly classified as fake;

The number of false positive examples is the number of news articles incorrectly classified as fake;

The number of true negative examples is the number of news articles, correctly classified as true;

The number of false negative examples is the number of news articles incorrectly classified as true;

The precision for the given classifier equals to 0.71; recall, on the other hand equals to

0.13. Such a low value of the recall, once again, is caused by the skewness of the data in the test dataset. We believe that precision is the most important characteristic of the given classifier.

News article type	Total number of news in test dataset	Number of correctly classified news	Classification accuracy
True	881	666	75.59%
Fake	46	33	71.73%
Total	927	699	75.40%

Figure 4.2: Analysis Table[4]

Chapter 5

Conclusion and Future Work

The results obtained above are very promising. This method demonstrates that term frequency is potentially predictive of fake news - an important first step toward using machine classification for identification. The best performing models by overall ROC AUC are Stochastic Gradient Descent models trained on the TF-IDF feature set only. We observe that PCFGs do not add much predictive value, but balance the Recall for our top performing model. This indicates that PCFGs are good for a Fake-News Filter type implementation versus, say, targeting fake news sites for review. TF-IDF shows promising potential predictive power, even when ignoring named entities, but we remain skeptical that this approach would be robust to changing news cycles. However, this would require a more complete corpus.

Despite the high performance of our classifier, there is definitely scope for improvement. We evaluated our models using absolute probability thresholds, which may not be the most reliable for models where probability scoring is not well-calibrated. While TF-IDF performs better, we are possibly over fitting to topics/terms important in the ongoing news cycle. Also, a vectorized approach like ours makes it technically hard to see which individual features are most important, thus hampering our analysis. These issues limit our analysis and thus prevent broader generalizability. We plan to address these issues in a future work.

References

[1] N. J. Conroy, V. L. Rubin, and Y. Chen, Automatic deception detection: Methods for finding fake news, Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 14, 2015."

[2] Shlok Gilda, Department of Computer Engineering, Evaluating Machine Learning Algorithms for Fake News Detection,2017 IEEE 15th Student Conference on Research and Development (SCOReD)."

[3] Hadeer Ahmed1, Issa Traore1, and Sherif Saad2 1 ECE Department, University of Victoria, Victoria, BC, Canada 2 School of Computer Science, University of Windsor, Windsor, ON, Canada "

[4] https://indianexpress.com/article/education/iiit-delhi-students-create-whatsfarzi-app-to-detect-fake-news-5599996/ "

[5] http://www.smdi.com/evolution-machine-learning "

[6] https://kastanday.com/fake-news-detection-details/machinelearningforstancedetection "

[7] https://www.pantechsolutions.net/fake-news-detection-using-machine-learning "

[8] https://www.csail.mit.edu/news/detecting-fake-news-its-source "