FAKE NEWS DETECTION SYSTEM USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING TECHNIQUES

by

RAMKUMAR V 2015103018
HARSHAVARDHANA M 2015103012

A project report submitted to the

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI – 25

SEPTEMBER 2018

BONAFIDE CERTIFICATE

Certified that this project report titled **FAKE NEWS DETECTION USING NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING TECHNIQUES** is the bonafide work of **RAMKUMAR V(2015103018)** and **HARSHAVARDHANA M (2015103012)** who carried out the project work for **Creative and Innovative Lab** under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

Place: Chennai

Date:

Bhuvaneshwari R
Teaching Fellow
Department of Computer Science and Engineering
Anna University, Chennai – 25

ABSTRACT

The term 'fake news' became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views. In this project, we seek to produce a model that can accurately predict the likelihood that a given article is fake news.

This project explores the application of natural language processing techniques and machine learning for the detection of 'fake news' that is, misleading news stories that come from non-reputable sources. Using a dataset obtained from Signal Media and a list of sources from OpenSources.co, we count vectorizer and apply term frequency-inverse document frequency (TF-IDF) of bi-grams to a corpus of about 11,000 articles. We test our dataset on Random Forest classifer to arrive at the solution of whether the particular news article is fake or not.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DESCRIPTION

Fake news, one of the biggest new-age problems has the potential to mould opinions and influence decisions. The proliferation of fake news on social media and Internet is deceiving people to an extent which needs to be stopped. Fake news itself is not a new problem, and the media ecology has been changing over time from newsprint to radio/television, and recently online news and social media. The impact of fake news on traditional media can be described from the perspective of psychology and social theories. For example, two major psychology factors make consumers naturally vulnerable to the fake news: (i) Naïve Realism: consumers tend to believe that their perceptions of reality are the only accurate views. (ii) Confirmation Bias: consumers prefer to receive information that confirms their existing views. The existing systems are inefficient in giving a precise statistical rating for any given news claim.

This system aims to develop a machine learning program to identify when news source may be producing fake news. We aim to use a corpus of labeled real and fake new articles to build a classifier that can make decisions about information based on the content from the corpus. The model will focus on identifying fake news, based on multiple articles originating from a source.

1.2 SCOPE

The scope of this projects includes various fields such as Social media where the fake news spreads like a rapid fire. This rapid spread of online misinformation poses an increasing risk to societies worldwide. It is a serious problem calling for solutions. This can be a new system of check and balance to prevent fake news spreading.
An algorithm-based system that identifies telltale linguistic cues in fake news stories could provide news aggregator and social media sites like Google News with a new weapon in the fight against misinformation. This system aims to use various NLP and classification techniques to help achieve maximum accuracy.

1.3 CONTRIBUTION

We used Random forest classifier and used Feature selection methods like Stemming-In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form – generally a written word form.Such technique helps to treat similar words (like “write” and “writing”) as the same words and may improve classifier’s performance as well. Stop words removal-Stop words are the words, that are common to all types of texts (such as articles in English). This words are so common, that they don’t really affect the correctness of the information in the news article, so it makes sense to get rid of them, Tokenization to improve the accuracy of prediction. Snowball stemmer is used to enhance the process of stemming.

In machine learning problems it is often the case when getting more data significantly improves the performance of a learning algorithm.The dataset, that was described in this article contains around 15359 articles. This would be of a better for the learning process.

1.4 SWOT ANALYSIS

1.4.1 Strength

- Access to reliable news information.
- Prevent the proliferation of fake news in social media and internet.
- Prevent misleading of agency ,person or rival.
- Prevent confusion and moral panic and eventually undermine an informed citizenry.

1.4.2 Weakness

- If internet connectivity is lost, then source verification is important.
- Difficulty in specifying the percentage of error and level of truthfulness.
- Large datas needed for more optimized prediction

1.4.3 Opportunity

- Redemption of agency, person or rival due to proliferation false news.
- Awareness among the public.
- This can be integrated with various social media.

1.4.4 Threat

Incorrect data can lead to incorrect prediction, thereby reducing the trustworthiness of the predictor.

1.5 PESTEL ANALYSIS

1.5.1 Political Factor

- Government laws and norms are factored in while designing a model which is ethical in all aspects.
- The proliferation of fake news will reduce significantly and the reputation of individual or company will be withheld.

1.5.2 Economic Factor

- A lot of defamation cases dude to falsified news can be reduced by using this method of fake news detection.
- To train large datasets for higher accuracy, we require high processing power systems.

1.5.3 Social Factor

This method of fake news detection helps in withholding the reputation of individual or organization in a society.

1.5.4 Technological Factor

- This model is based upon on of the most commonly used and highly accurate machine learning classifier namely Random Forest Classifier.
- Future works include continuous feeding of news for better performance and efficiency.

1.5.5 Legal Factor

Due to governmental laws, it is sometimes difficult for implementing machine learning models for commercial use.

CHAPTER 2

RELATED WORK

2.1 FAKE NEWS DETECTION USING NAIVE BAYES CLASSIFIER

In this the Naive Bayes classifier was specifically used for fake news detection; also, the developed system was tested on a relatively new data set, which gave an opportunity to evaluate its performance on a recent data.The dataset used was Facebook Newsfeed dataset.

Naive Bayes classifiers are a popular statistical technique of email filtering. 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.

Formula for calculating the conditional probability

$$Pr(F|W) = Pr(W|F)\cdot Pr(F)/(Pr(W|F)\cdot Pr(F)+Pr(W|T)\cdot Pr(T)),$$

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

The probability of finding specific word in fake news article as a ratio of the fake news articles, that contain this word to the total number of fake news articles.
This formula is often used for spam filtering.

The process involves

1. Article Information Binding
2. Article filtering based on the presence of content and relevant labels
3. Separating data in training, testing and validation sets
4. Training the Naïve Bayes classifier
5. Testing and accuracy evaluation

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% of it are fake news.

Improvements Suggested:

- Get more data and use it for training
- Use the dataset with much greater length of the news articles
- Remove stop words from the news articles
- Use stemming
- Treat rare words separately

2.2 FAKE NEWS DETECTION USING STACKED ENSEMBLE OF CLASSIFIERS

The Fake news Detection is implemented as stance classification task. Fake news challenge is a text classification task: given a headline and article body - the classifier must first predict whether the two are related and if so, must then further assign a stance label - whether the headline agrees with, disagrees with or is discussed by (observing) the article.

An additional Blind test set was used.

The Ensemble Of Classifiers include(**C1-C5**):

C1: Concatenate average word2vec vectors for headline and article body, cosine similarity between headline and article body tf-idf vectors and counts of refuting words. 4-way classification using a (300,8) multi-layer perceptron (MLP) with ReLU activation function.

C2: Averageword2vecembeddingsforheadline words and article words excluding stop words, indicator features for punctuation, word overlap, counts of refuting words using (1010,6) MLP.

C3: 4-way classification using one-vs-all logistic regression

C4: Concatenate word2vec embedding for headline and article words. 4-way classification using (256,128,128) MLP.

C5: Official FNC baseline classifier

CM: Gradient boosted decision tree classifier using as features the values predicted from C1-C5 and all the features from the FNC baseline classifier.

The master meta-classifier in our entry leverages additional information about which slave predictions to favor given a certain headline and article pair.

Cross-validating against the development split yielded classifiers that were not able to generalize to the unseen articles in the test set, harming the classification accuracy

One factor limiting the ability our model(s) to generalize is the overlap of headlines between the training and development evaluation dataset.

Future evaluations:

Consider temporal splits, i.e. deriving training, development and test sets from articles from different periods, which would also mimic to an extent how these models might be used in practice.

2.3 FAKE NEWS DETECTION USING NLP AND CLASSIFICATION TECHNIQUES

Here The URL of the article that the user wants to authenticate, after which the text is extracted from the URL. The extracted text is then passed on to the data preprocessing unit. The data preprocessing unit consists of various processes like the Tokenization and Generation of the word cloud. The outputs from these processes play an important role in further analyzing the data.

The core deciding factors Stance detection and measuring document similarity.Stance is a mental or an emotional position adopted by the author with respect to something. Stance detection is an important part if NLP and has wide applications. The stance of the author can be divided into various categories like Agree, Disagree, Neutral or Unrelated with respect to the title. Giving each of these categories weights can help us in the final conclusion of whether a news article is fake or not.

The second method is to use document similarity or if-idf to know how similar a document is to top search results. This too can give us an insight into the authenticity of a news article.

- Extraction of data(Web Crawler)
- Tokenization & Generation of wordcloud
- Stance Detection
- Document Similarity
- Classification algorithm & F-Score Generation

2.4 FAKE NEWS DETECTION USING LINGUISTIC APPROACH AND NETWORK

The simplest method of representing texts is the “bag of words” approach, which regards each word as a single, equally significant unit.

In the bag of words approach, individual words or “n-grams” (multiword) frequencies are aggregated and analyzed to reveal cues of deception.

2.4.1 Syntax Analysis

Deeper language structures (syntax) have been analyzed to predict instances of deception. Deep syntax analysis is implemented through **Probability Context Free Grammars (PCFG)**. Sentences are transformed to a set of rewrite rules (a parse tree) to describe syntax structure, for example noun and verb phrases, which are in turn rewritten by their syntactic constituent parts.

Third-party tools, such as the Stanford Parser, AutoSlog-TS syntax analyzer and others assist in the automation.

2.4.2 Semantic Analysis

The intuition is that a deceptive writer with no experience with an event or object (e.g., never visited the hotel in question) may include contradictions or omission of facts present in profiles on similar topics.

Extracted content from key words consists of *attribute:descriptor* pair.

This model utilized the n-gram plus syntax model.

2.4.3 Limitations

The ability to determine alignment between attributes and descriptors depends on a sufficient amount of mined content for profiles, and the challenge of correctly associating descriptors with extracted attributes.

2.4.4 Centering Resonance Analysis (CRA)

Centering resonance analysis (CRA), a mode of network-based text analysis, represents the content of large sets of texts by identifying the most important words that link other words in the network.

Combining sentiment and behaviour studies have demonstrated the contention that sentiment-focused reviews from singleton contributors significantly affects online ranking and that this is an indicator of “shilling” or contributing fake reviews to artificially distort a ranking.

Improvements:

- Linguistic processing should be built on multiple layers from word/lexical analysis to highest discourse-level analysis for maximum performance.
- As a viable alternative to strictly content-based approaches, network behavior should be combined to incorporate the ‘trust’ dimension by identifying credible sources.
- Tools should be designed to augment human judgement, not replace it. Relations between machine output and methods should be transparent.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The system outputs Fake News or Not for a given news sentence in English as an input. The output sentence should adhere to the following requirements:

- The system must be able to correctly predict if a given news is Fake or Not
- The output should contain the confusion matrix generated while testing the dataset
- The system must be optimized for time and space complexities`

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 User Interface

There must be a simple and easy to use user interface where the user should be able to enter his input sentence(s). The intermediate result and also the output should be displayed in the screen.

3.2.2 Hardware

No special hardware interface is required for the successful implementation of the system.

3.2.3 Software

- Operating System: Linux
- Programming Language: Python

- Dataset:Liar Dataset
- Tools: Sublime Text Editor

3.2.4 Performance

The system must be optimized, reliable, consistent and available all the time.

3.3 CONSTRAINTS AND ASSUMPTIONS

3.3.1 Constraints

- The system would work only for those words in the dictionary. There are around 1,00,000 English words.
- Training the dataset which doesn't have proper values will lead to incorrect training of the dataset which will result in incorrect output. However, this can be eliminated by cleaning the dataset by eliminating the dataset which doesn't have proper values.
- The accuracy of the classifier depends on the correctness of dataset. If the dataset is not correct, it will bring down the accuracy of the classifier.

3.3.2 Assumptions

- The input sentence is assumed to be grammatically correct.
- The input is assumed to be either an assertive sentence or an imperative sentence.
- The input sentence does not have any spelling mistakes.

3.4 SYSTEM MODELS

3.4.1 Use Case Diagram

The overall usecase diagram of the entire system is shown in figure 3.1.

Pre condition: A news article is given as input by the user.

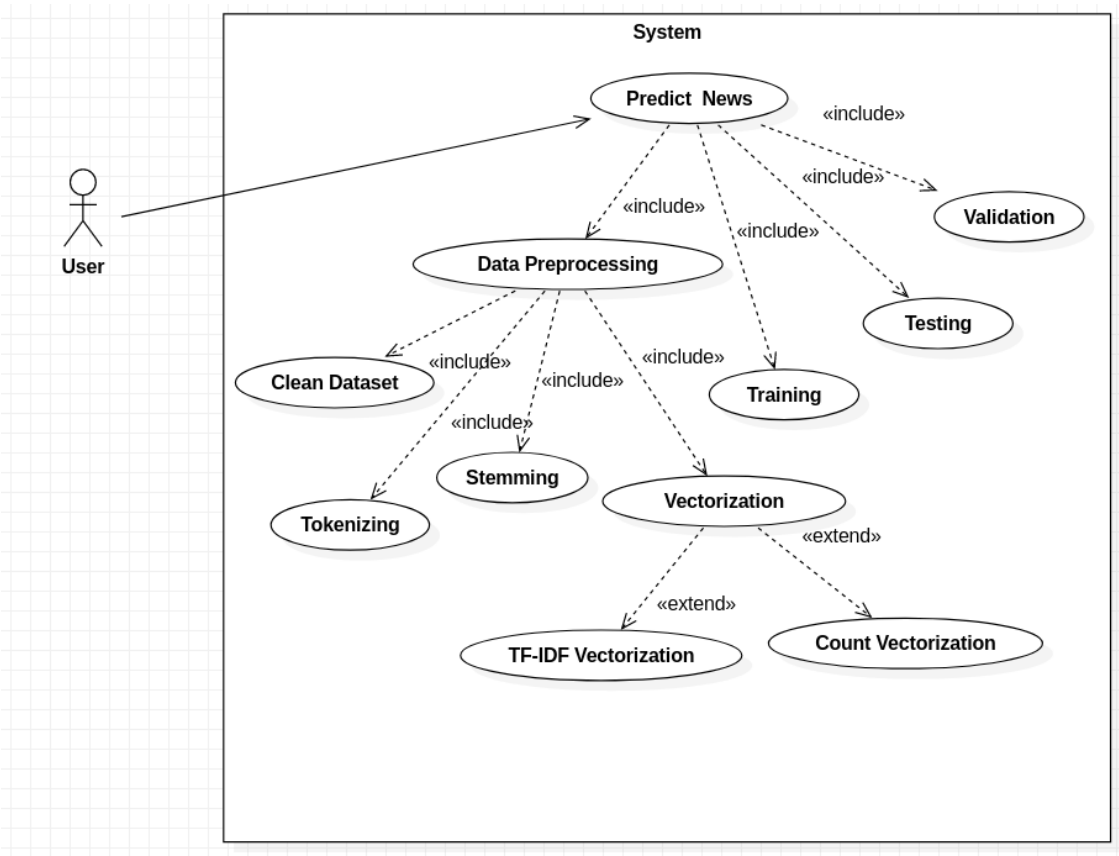


Figure 3.1 Overall Use Case Diagram

Post condition: The predicted result whether the news article is fake or not.

3.4.2 Sequence Diagram

The sequence of steps involved in the process are shown in figure 3.2.

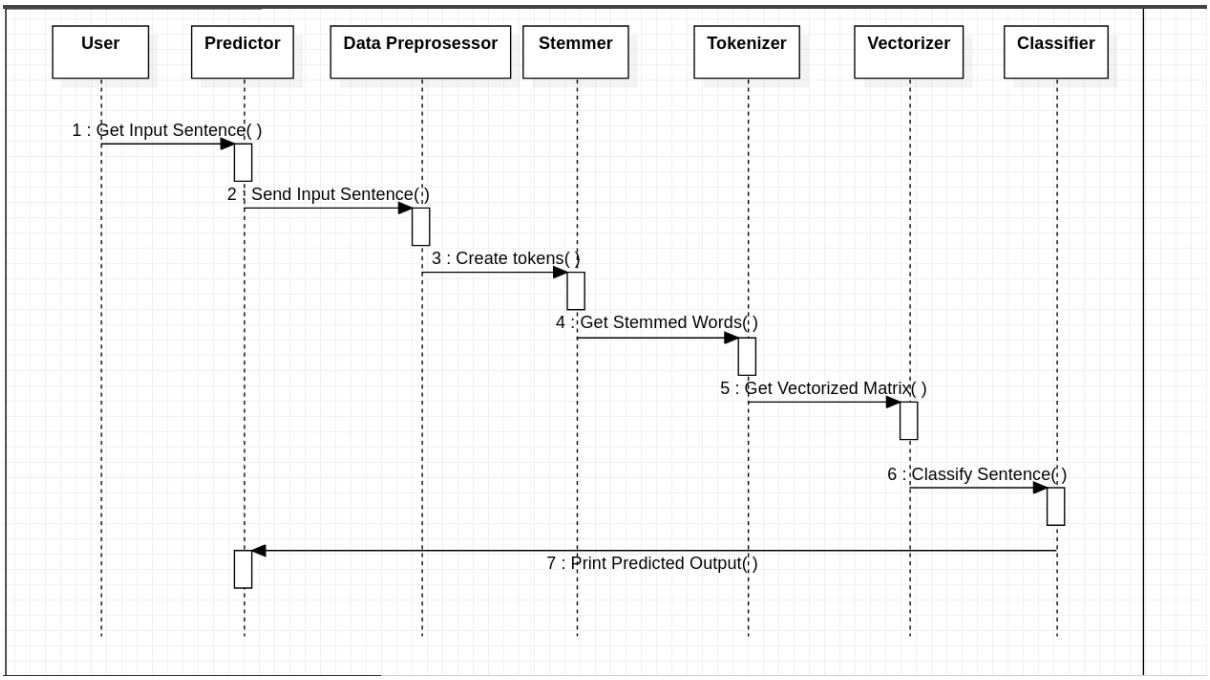


Figure 3.1 Sequence Diagram

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The dataset used for Fake News detection is LIAR dataset which is a benchmark dataset for Fake news detection. The input dataset is first given to the data processing unit where the dataset is checked for no of entries and also null entries in the database.After which a distribution is created to separate the dataset for training,testing and validation phase.Afer this processs the dataset is tokenied using Standard nltk tokenizer. Tokenizing means splitting your text into minimal meaningful units. It is a mandatory step before any kind of processing.The basic tokenizer (like in NLTK) will split your text into sentences and your sentences into typographic tokens.After tokenization,stop words are removed from the tokenized set as they wont contribute much as a feature. Stop words are natural language words which have very little meaning, such as "and", "the", "a", "an", and similar words. After stop word removal,Stemming is performed. Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP). Stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. After Stemming,we will be having the desired features for classification.

After data processing,feature extraction comes into picture,Count vectorizer is used to build to term-document matrix which is later used for

tf-idf feature extraction.After the feature extraction phase, the features are fed into the Random forest classifier.The model is trained,tested and validated. The confusion matrix and accuracy of the final model are generated.

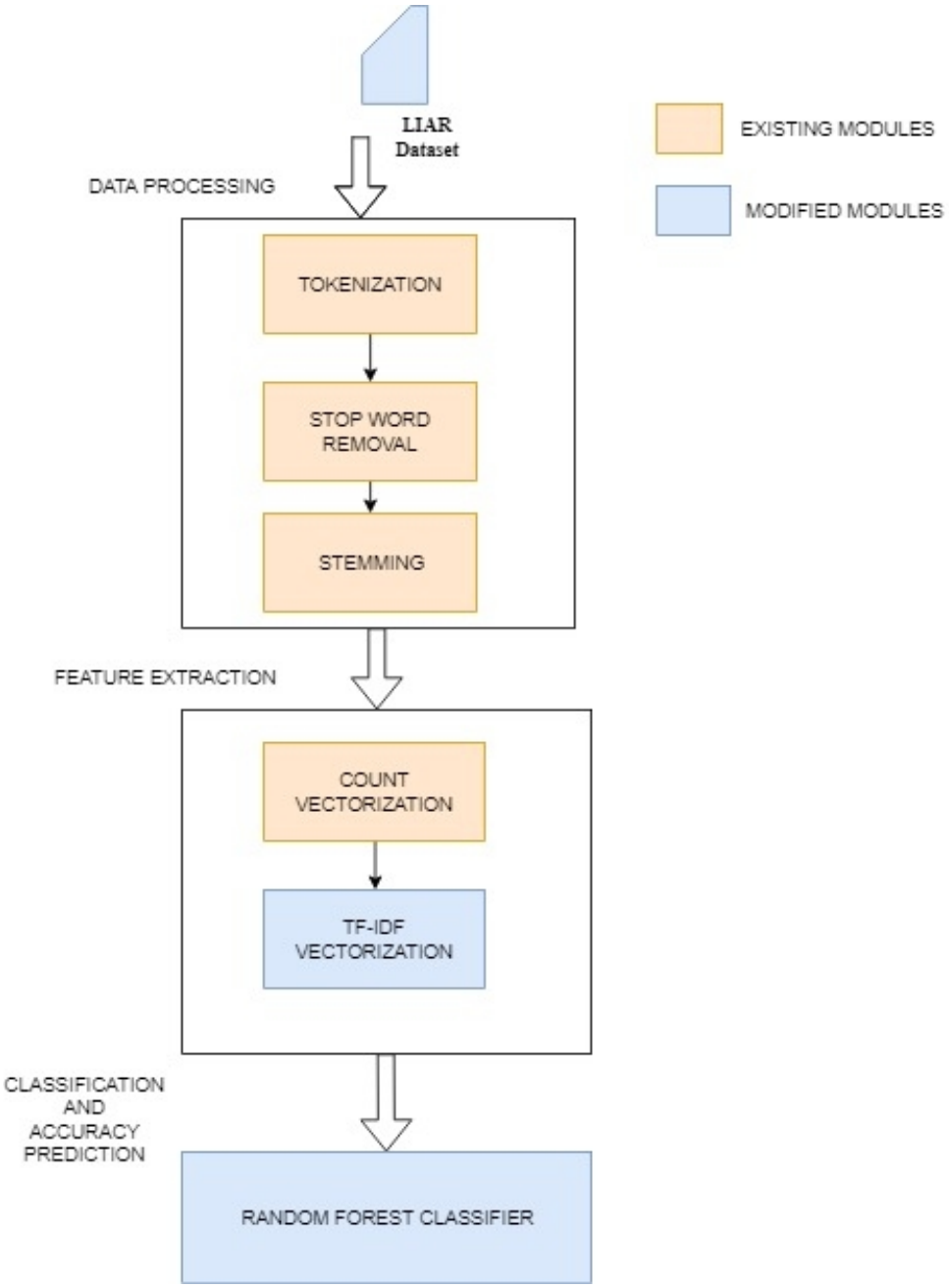


Figure 4.1 System Architecture

4.2 MODULE DESIGN

4.2.1 Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens* , perhaps at the same time throwing away certain characters, such as punctuation.

Eg. Input: Friends, Romans, Countrymen, lend me your ears; Output:

The tokenizer's features will depend on what you want to do next. If you want to parse the text, you will want it as clean as possible (hence a lot of modifications) but if you want to train a word2vec kind of model, the basic will be enough (because of the size of your corpus and the nature of the algorithm).

It is implemented using *nltk*.

4.2.2 Stopword Removal

Stop words are words which are filtered out before or after processing of natural language data (text). A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Stop word removal is performed using the *nltk.corpus* module using the *Stopwords* package.

4.2.3 Stemming

Stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word.

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

There are different algorithms that can be used in the stemming process, but the most common in English is Porter stemmer. The rules contained in this algorithm are divided in five different phases numbered from 1 to 5. The purpose of these rules is to reduce the words to the root.

Here we implemented the stemming process with *Snowball stemmer.* The difference between a porter and a snowball stemmer is that it has slightly faster computation time than porter, with a fairly large community around it.The implementation is done with *nltk.stem.porter module* using *PorterStemmer* package.

4.2.4 Count Vectorizer

It converts the text document into numerical feature vectors.

The better way is

- Assign a fixed integer id to each word occurring in any document of the training set (for instance by building a dictionary from words to integer indices).
- For each document #i, count the number of occurrences of each word w and store it in X[i, j] as the value of feature #j where j is the index of word w in the dictionary

Convert a collection of text documents to a matrix of token counts

This implementation produces a sparse representation of the counts using scipy.sparse.csr_matrix.

This can be done by assigning each word a unique number. Then any document we see can be encoded as a fixed-length vector with the length of the vocabulary of known words. The value in each position in the vector could be filled with a count or frequency of each word in the encoded document.This is the bag of words model, where we are only concerned with encoding schemes that represent what words are present or the degree to which they are present in encoded documents without any information about order.

It is implemented using *sklearn.feature_extraction.text module.*

4.2.5 TF-IDF Vectorizer

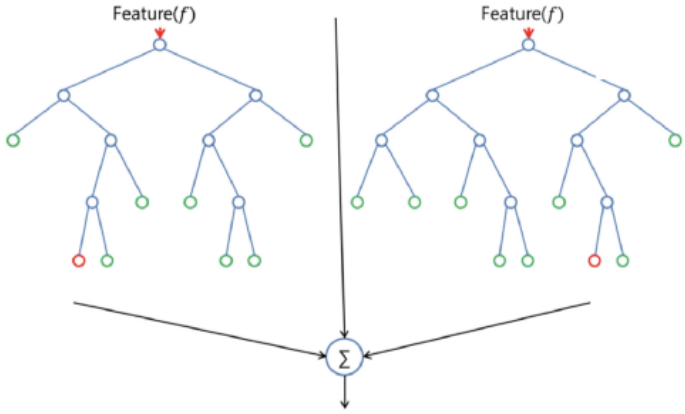
A Term Frequency is a count of how many times a word occurs in a given document (synonymous with bag of words). The Inverse Document Frequency is the the number of times a word occurs in a corpus of documents. tf-idf is used to weight words according to how important they are. Words that are used frequently in many documents will have a lower weighting while infrequent ones will have a higher weighting.

The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

It is implemented with *sklearn.feature_extraction.text module using TfidfTransformer.*

4.2.6 Random Forest Algorithm

Random Forest is a supervised learning algorithm. Like you can already see from it's name, it creates a forest and makes it somehow random. The "forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.



CHAPTER 5

SYSTEM DEVELOPMENT

The system described consists of various packages like nltk, numpy,seaborn, scikit-learn, etc. Figure 5.1 shows the packages used.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.snowball import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
import seaborn as sb
from sklearn.feature_extraction.text import TfidfTransformer
```

Figure 5.1 Code Overview

An overview of the algorithm of entire system is shown below. The dataset is cleaned first by removing the null values. It is then tokenized and Stemmed. The count vector is then generated using the array of statements containing stemmed words. The output of the count vector is then given to tf-idf transform which gives the tf-idf representation.

5.1 PROTOTYPE ACROSS THE MODULES

The input and output to each module of the system is described in this section.

- **Dataset cleaning:** This module checks for null values in the dataset and removes the null value, if any, and returns the cleaned dataset.
- **Tokenizing:** This module seperates each words as tokens in a statement and removes all the stopwords and returns an array consisting of tokens.
- **Stemming:** This module takes each tokenized words and returns the root word.
- **Count Vector:** This module takes the stemmed words and keeps count of number of occurances of each word in a statement. It is returned in form of an array
- **TF-IDF Vectorizer:** This module takes the count vector as input and returns the tf-idf transformed array.
- **Random Forest Classifier:** This module builds several decision trees to form random forest tree using tf-idf transformed array as input. This decision trees are then used to predict if a given statement is a fake news or not.
- **Performance Evaluation:** This module uses confusion matrix and a test dataset with actual label. This testing dataset is feeded to random forest and it predicts the output. The actual label and label of predicted output is checked. If the actual label and predicted label is true, it is known as true positive. If the actual label and predicted label is false, it is known as true negative. If the actual label is true and predicted label is false, it is known as false negative. If the actual label is false and predicted label is true, it is known as false positive. These values can be used to check the accuracy of the system.

5.2 RANDOM FOREST ALGORITHM

The Random forest algorithm is given below.

RANDOM FOREST(TF-IDF VECTOR)

1. Randomly select “k” features from total “m” features.
    1. where k << m
2. Among the “k” features, calculate the node “d” using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “l” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

5.3 DEPLOYMENT DETAILS

The deployment of the system requires python 2.7, nltk stopword corpus and packages like scikit-learn, numpy, seaborn, nltk, pandas, mathplot lib, etc. The liar dataset should be available to train the random forest classifier. Any IDE that supports Python 2.7 can be used.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 DATASET FOR TESTING

The input to the system consists of 15,400 news headlines taken from liar dataset. The test data consists of sentence and label. The sentence is an english statement which is the headlines of a news. The label says if the news is fake or not. The label is a boolean value, it is true if the news is true else false. The result of the module testing as well as testing of entire system are summerized below.

6.2 OUTPUT OBTAINED IN VARIOUS STAGES

This section shows the results obtained during module testing.

6.2.1 Input Sentence

The input sentence shown in figure 6.1 was given to the system.

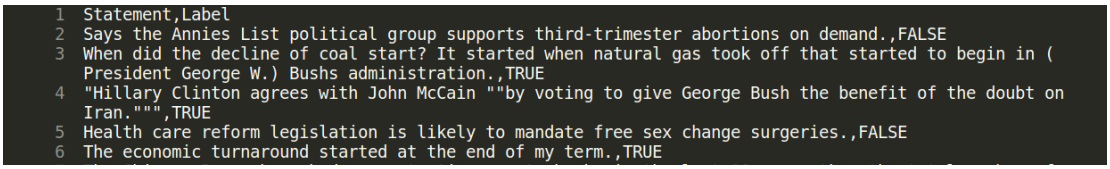


Figure 6.1 Input Sentence

6.2.2 Dataset cleaning

The output of dataset cleaning is shown in figure 6.2.

```
training dataset size:
(15359, 2)
                                                Statement  Label
0  Says the Annies List political group supports ...  False
1  When did the decline of coal start? It started...   True
2  Hillary Clinton agrees with John McCain "by vo...   True
3  Health care reform legislation is likely to ma...  False
4  The economic turnaround started at the end of ...   True
5  The Chicago Bears have had more starting quart...   True
6  Jim Dunnam has not lived in the district he re...  False
7  I'm the only person on this stage who has work...   True
8  However, it took $19.5 million in Oregon Lotte...   True
9  Says GOP primary opponents Glenn Grothman and ...   True
Checking data qualitites...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15359 entries, 0 to 15358
Data columns (total 2 columns):
Statement    15359 non-null object
Label        15359 non-null bool
dtypes: bool(1), object(1)
memory usage: 135.1+ KB
check finished.
```

Figure 6.2 Output of dataset cleaning

6.2.3 Tokenization and Stemming

The output of tokenization and stemming is shown in figure 6.3.

```
[u'say anni list polit group support third trimest abort demand', u'declin coal start start natur g
as took start begin presid georg w bush administr', u'hillari clinton agre john mccain vote give ge
org bush benefit doubt iran', u'health care reform legisl like mandat free sex chang surgeri', u'ec
onom turnaround start end term', u'chicago bear start quarterback last year total number tenur uw f
aculti fire last two decad', u'jim dunnam live district repres year', u'person stage work activ las
t year pass along russ feingold toughest ethic reform sinc waterg', u'howev took million oregon lot
teri fund port newport eventu land new noaa marin oper center pacif', u'say gop primari oppon glenn
 grothman joe leibham cast compromis vote cost million higher electr cost', u'first time histori sh
are nation popular vote margin smaller latino vote margin', u'sinc near million american slip middl
 class poverti', u'mitt romney governor massachusett didnt slow rate growth govern actual cut', u'e
conomi bled billion due govern shutdown', u'afford care act alreadi sens waiv otherwis suspend', u'
last elect novemb percent american peopl chose vote percent young peopl percent low incom worker ch
ose vote', u'mccain oppos requir govern buy american made motorcycl said buy american provis quot d
isgrac', u'u rep ron kind wis fellow democrat went spend spree credit card max', u'water rate manil
a philippin rais percent subsidiari world bank becam partial owner', u'almost peopl left puerto ric
o last year', u'women men make less adjust inflat john kitzhab first elect governor', u'unit state
highest corpor tax rate free world', u'best year auto industri america histori', u'say scott walker
 favor cut famili children health care', u'say mitt romney want get rid plan parenthood', u'dont kn
ow jonathan gruber', u'hate crime american muslim mosqu tripl pari san bernardino', u'rick perri ne
ver lost elect remain person texa governorship three time landslid elect', u'isi support tweet shoo
t began chattanooga tenn', u'youth unemploy minor communiti percent']
```

Figure 6.3 Output of Morphological Analyser

6.2.4 TF-IDF Vectorizer

The output consists of a array consisting of TF-IDF transformation. The output is shown in figure 6.4.

```
(0, 7667)     0.452280928487
(0, 7482)     0.269432046292
(0, 7234)     0.205936770991
(0, 6457)     0.115280231654
(0, 5658)     0.28605471857
(0, 4315)     0.311877173657
(0, 3178)     0.284653501818
(0, 1904)     0.349911912302
(0, 304)      0.471501751342
(0, 22)       0.249548214281
(1, 7560)     0.202949161519
(1, 7025)     0.681921267963
(1, 5766)     0.149163735313
(1, 4931)     0.27594596766
(1, 3004)     0.223652618469
(1, 2962)     0.227950946213
(1, 1854)     0.274848642839
(1, 1386)     0.269782754198
(1, 1007)     0.197833301793
(1, 662)      0.259054781413
(1, 93)       0.197091327833
(2, 8031)     0.192828353822
(2, 4565)     0.272876016685
(2, 3913)     0.262848147454
```

Figure 6.4 Output of TF-IDF Vectorizer

6.2.5 Random Forest Prediction (Testing)

The output after prediction of testing dataset is shown in figure 6.5.

```
[False False  True ..., False  True  True]
```

Figure 6.5 Output of Testing

6.2.6 Confusion matrix and accuracy

The generated confusion matrix's output and accuracy is shown in figure 6.6. The confusion matrix is generated by checking how many dataset has been classified correctly and how many have been classified wrongly. Accuracy is predicted using the values in confusion matrix.

```
[[ 909  486]
 [ 484 1193]]
Accuracy: 68.4244791667
Precision: 0.710541989279
Recall: 0.711389385808
F-Score: 0.710965435042
```

Figure 6.6 Output of Confusion Matrix

6.3 PERFORMANCE EVALUATION

The performance of the entire system is evaluated using the standard parameters described below.

6.3.1 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Let's now define the most basic terms, which are whole numbers:

- True positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
- True negatives (TN): We predicted no, and they don't have the disease.
- False positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- False negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

The above terms can be used to find the accuracy of the system. The accuracy is found using the following formula:

$$\text{Accuracy} = \frac{(\text{True Positive} + \text{True Negative})}{(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})} \text{ X } 100\%$$

The confusion matrix and accuracy of the system is shown in figure 6.7.

```
[[ 909  486]
 [ 484 1193]]
Accuracy: 68.4244791667
Precision: 0.710541989279
Recall: 0.711389385808
F-Score: 0.710965435042
```

Figure 6.7 Confusion Matrix and Accuracy of the system

CHAPTER 7

CONCLUSIONS

7.1 SUMMARY

The developed Random forest classifier classifies the news articles between real and fake. The input dataset is the *LIAR* dataset which contains 15,000 news articles for classification. Data processing steps involves tokenization, stopword removal, stemming and then count vectorization whose result is fed into the tf-idf filtration. We separate the news article statements into tokens by using nltk tokenizer and then remove the stop words in that using the nltk stopword removal. Then stemming is performed to derive the root word for important features which are used in the formation of decision trees. Then count vectorization is performed that develops the term-document matrix which contains the matrix of unique words in the dataset and their frequencies. The ouput of the count vectorization is fed into the tf-idf vector which gives the term frequency and inverse document frequency for the input dataset. Then the selected features are fed into Random Forest Classifier which constructs multiple decision trees and the output of which depends on the majority of the result by decision trees. Then after testing and validation phase confusion matrix is generated and accuracy of prediction is measured.

New methods like Stopword removal,Stemming,tf-idf filtration are used for classification which are suggested in the referenced papers.The dataset has been increased to improve the accuracy of classification.

7.2 CRITICISMS

Even though the suggested methods have been employed the accuracy showed only a sight variation. This may be due to the size of the dataset. Dataset has to be increased to improve the accuracy of predication. Ensemble methods can be tried to check for the classification. Improved methods have to be employed inorder to improve the accuracy further.

7.3 FUTURE WORK

Apart from stemming,Count vectorization and tf-idf vector generation new improved methods like Tensor Embeddings and Label Propagation can be employed. Feature extraction can be improved by employing new methods or by modifying the existing ones.The size of the dataset can be increased. Various algorithms can be employed for the process of classification.

REFERENCES

1. Steve Fuller [online] *Statistics & Facts about Fake News* Availablehttps://www.statista.com/topics/3251/fake-news/

2. Natasha Lomas[online] *Fake news is an existential crisis for social media* Available: https:// techcrunch.com/2018/02/18/fake-news-is-an-existential- crisis-for -social- media/

3. Katie Langin[online]*Fake news spreads faster than true news on Twitter—thanks to people, notbots* Available:http:// www.sciencemag.org/news/2018/03/fake-news-spreads-faster- true-news-twitter-thanks-people-not-bot

4. Mykhailo Granik, Volodymyr Mesyura ,*Fake News Detection Using Naive Bayes Classifier* [online]

5. JamesThorne ,MingjieChen ,GiorgosMyrianthous ,JiashuPu ,XiaoxuanWang, AndreasVlachos , *FakeNewsDetectionusingStackedEnsembleofClassifiers* [online]

6. Manisha Gahirwal,Sanjana, MogheTanvi ,KulkarniDevansh, KhakharJayesh Bhatia, *Fake News Detection* [online]

7. Niall J. Conroy, Victoria L. Rubin, and Yimin Chen , *Automatic Deception Detection: Methods for Finding Fake News* [online] Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pra2.2015.145052010082

8. Ver´onica P´erez-Rosas1, Bennett Kleinberg2, Alexandra Lefevre1 Rada Mihalcea1, *Automatic Detection of Fake News*[online]

9. Matthew N O Sadiku, Tochukwu P Eze, and Sarhan M Musa , *FAKE NEWS AND MISINFORMATION* [online] Available: https://ijasre.net/uploads/1/3629_pdf.pdf