Shahjalal University of Science and Technology

Department of Computer Science and Engineering

CSE 400

Fake News Detection in Social Media Using Machine Learning

Students

Arnab Sen Sharma
Reg. No.: 2013331017
4th year, 1st Semester

Maruf Ahmed Mridul
Reg. No.: 2013331015
4th year, 1st Semester

Department of Computer Science and Engineering

Supervisor

Md Saiful Islam
Assistant Professor
Department of Computer Science and Engineering

18th March, 2018

Fake News Detection in Social Media Using Machine Learning

A Thesis submitted to the Department of Computer Science and Engineering, Shahjalal University of Science and Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

Students

Arnab Sen Sharma	Maruf Ahmed Mridul
Reg. No.: 2013331017	Reg. No.: 2013331015
4^{th} year, 1^{st} Semester	4^{th} year, 1^{st} Semester

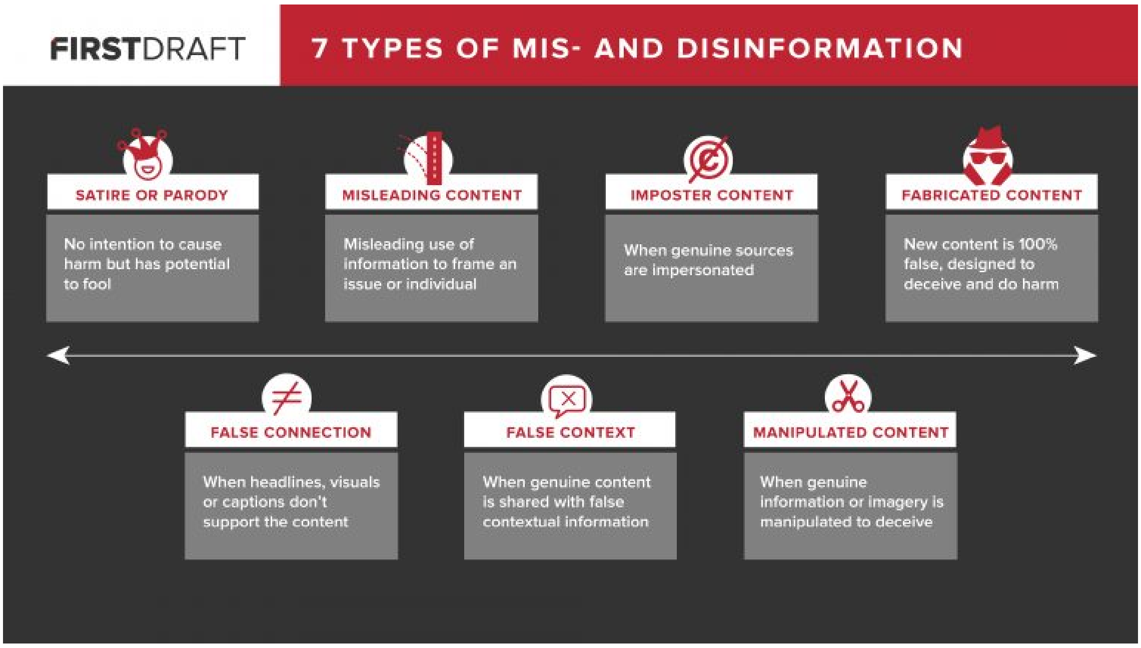
Department of Computer Science and Engineering

Supervisor

Md Saiful Islam
Assistant Professor
Department of Computer Science and Engineering

18th March, 2018

First Draft News defines a news fake if it has at least one of the following types of mis and disinformation.



FIRSTDRAFT

7 TYPES OF MIS- AND DISINFORMATION

SATIRE OR PARODY

No intention to cause harm but has potential to fool

MISLEADING CONTENT

Misleading use of information to frame an issue or individual

IMPOSTER CONTENT

When genuine sources are impersonated

FABRICATED CONTENT

New content is 100% false, designed to deceive and do harm

FALSE CONNECTION

When headlines, visuals or captions don't support the content

FALSE CONTEXT

When genuine content is shared with false contextual information

MANIPULATED CONTENT

When genuine information or imagery is manipulated to deceive

Figure 1.1: 7 types of mis and disinformation[9]

1.2 Problem Statement

Generally humans are not effective enough to detect whether a news is fake or not (DePaulo, Charlton, Cooper, Lindsay, & Muhlenbruck, 1997; Rubin & Conroy, 2012; Vrij, Mann, & Leal, 2012). There are actually three reasons for this. First, most people believe what they know and what information they receive is true. Second, some people easily accept fuzzy facts. Third, confirmation bias can cause people to see only what they want to see. So, it is very easy for fake news to deceive an average human being. Now, the deceived person can share, retweet those through various social media platforms and thus unintentionally play a part in the propagation of fake news. As a result, sometimes those spread virally. For example, a fake news during the Malaysian airlines incident titled *"MH370 found in near Bermuda Triangle"* became viral in twitter very fast. So, we really need an efficient system for fact-checking.

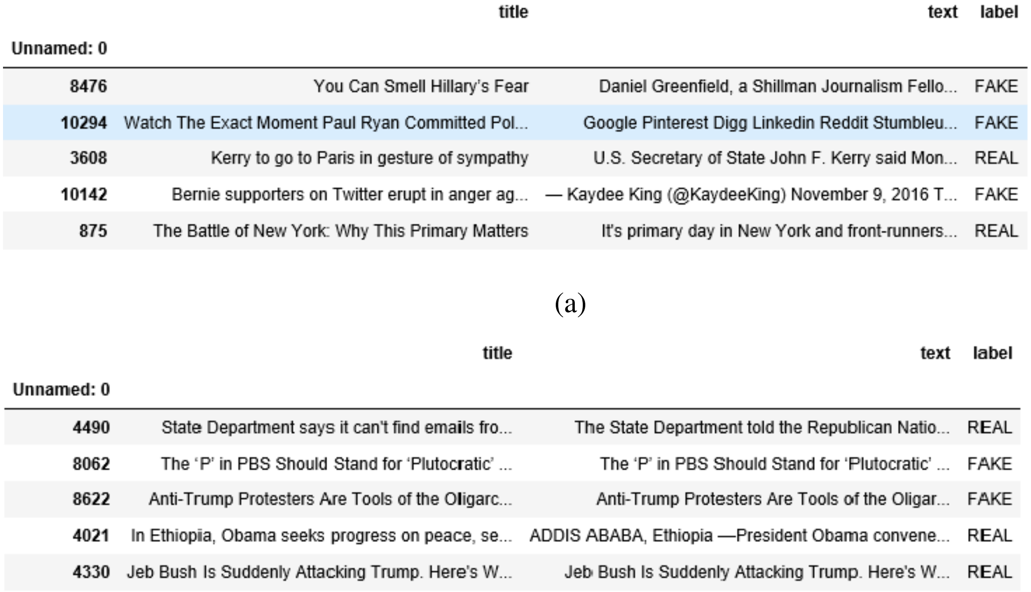
As a matter of fact, there are some web applications such as Snopes.com, FactCheck.org, PolitiFact etc which act as fact-checkers. But, these services use human staffs to manually check facts. Though these services provide accurate information most of the time, these are not efficient enough for the battle against fake news since they are not automated.

Chapter 3

Data Collection and Preprocessing

3.1 Data Collection

We have not actually collected any dataset ourselves. For experimenting what we have learned so far we have used an open dataset available on GitHub. The dataset is available at [17]. This dataset contains 6335 news articles labeled either 'FAKE' or 'REAL'. These news articles were collected during the time of presidential election of USA,2016. We are really grateful to **GeorgeMcIntire**[18] for such wonderful dataset, which really helped us get started. We used this dataset for training and testing our model.



Unnamed: 0	title	text	label
8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

(a)

Unnamed: 0	title	text	label
4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

(b)

Figure 3.3: An Overview of the Dataset

4.3 Feature Extraction

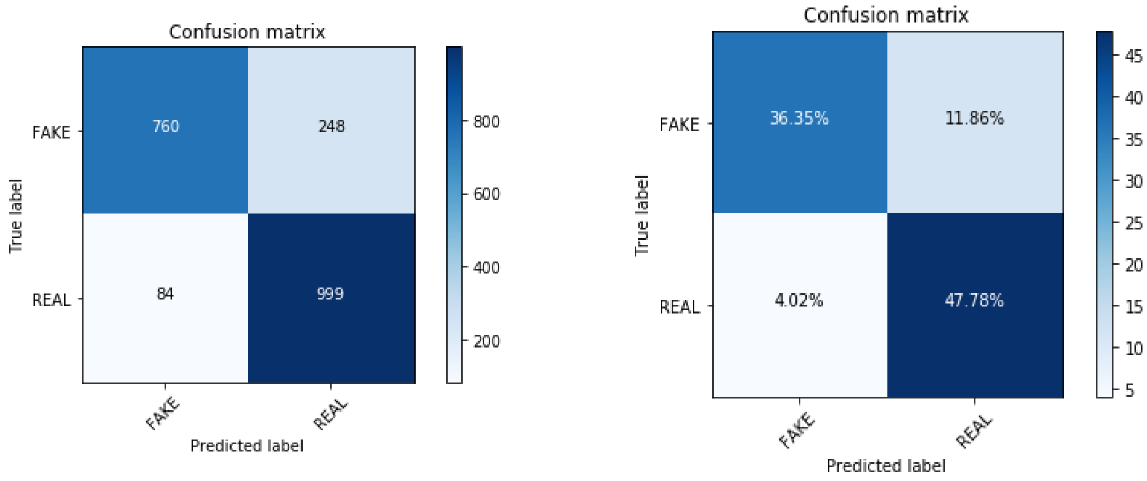
For now we have used the most common word embedding technique TF-IDF. We used *TfidfVectorizer* of sklearn for this technique. We set max_df = 0.6 to ignore the word which appear in more than 60% of the datasets. For now we use only article text as our dataset and discard all other information like source of the news.

4.4 Models and Evaluation

We have implemented several models using sklearn. We split our dataset in two parts using sklearn's test_train_split such that two third of the dataset was used as training data and rest one third was used as primary evaluation dataset. We have implemented Naive Bayes Tree , SVM , Passive Aggressive Classifier and Random Forest Classifier.

4.4.1 Naive Bayes classifier

We used Multinomial naive Bayes as our naive Bayes classifier. We trained this classifier using our training data and evaluated the classifier with our primary evaluation data which we got by splitting our dataset. This classifier gave us the evaluation score of 84.1%. Below is shown the confusion matrix on the classifiers performance on our primary evaluation dataset.



(a) Value

(b) Percentile

Figure 4.3: Confusion Matrix of Naive Bayes Classifier

Multinomial Naive Bayes has a parameter α which acts as a additive smoothing parameter. Tuning this α parameter we were able to achieve upto 86.861% accuracy. Below is shown different accuracy rates against different α values which we were able to achieve for this dataset.

```
Alpha: 0.00 Accuracy: 0.84422
Alpha: 0.10 Accuracy: 0.86861
Alpha: 0.20 Accuracy: 0.86000
Alpha: 0.30 Accuracy: 0.85761
Alpha: 0.40 Accuracy: 0.85331
Alpha: 0.50 Accuracy: 0.84901
Alpha: 0.60 Accuracy: 0.84757
Alpha: 0.70 Accuracy: 0.84422
Alpha: 0.80 Accuracy: 0.83801
Alpha: 0.90 Accuracy: 0.83227
Alpha: 1.00 Accuracy: 0.82748
Alpha: 1.10 Accuracy: 0.82509
Alpha: 1.20 Accuracy: 0.82414
Alpha: 1.30 Accuracy: 0.82127
Alpha: 1.40 Accuracy: 0.81935
```

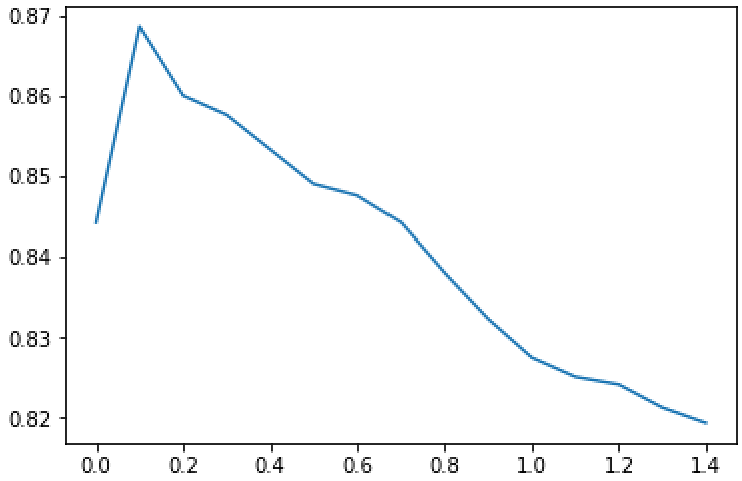
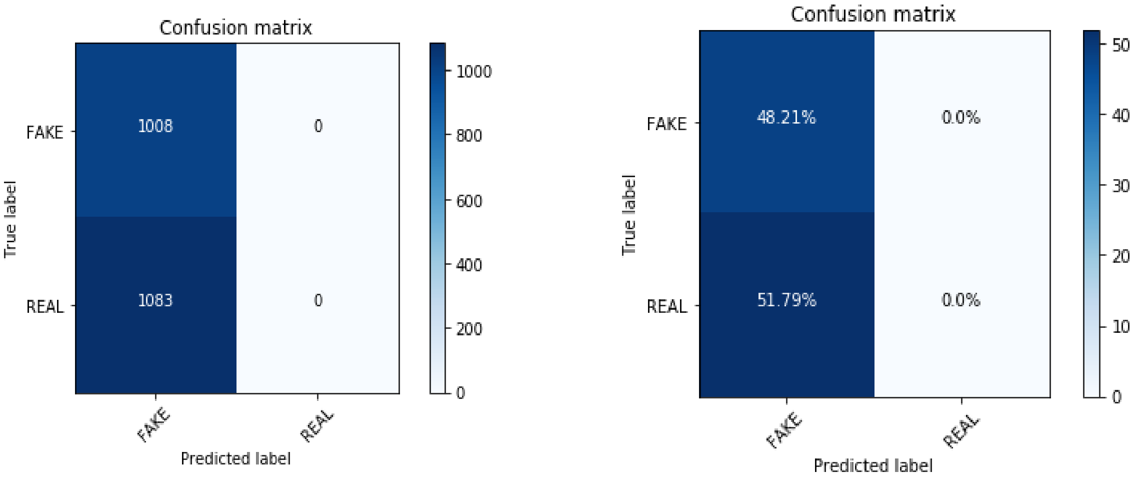


Figure 4.4: Different accuracy rates against different α values

4.4.2 SVM

Now we tried our luck with SVM in the hope of getting a better result. This model took almost 20 minutes to build, almost 10 times the time needed to build the naive Bayes model. But we were very surprised at first to see that during testing all of the news were classified as 'FAKE' and thus achieving a very poor accuracy rate 48.2%.

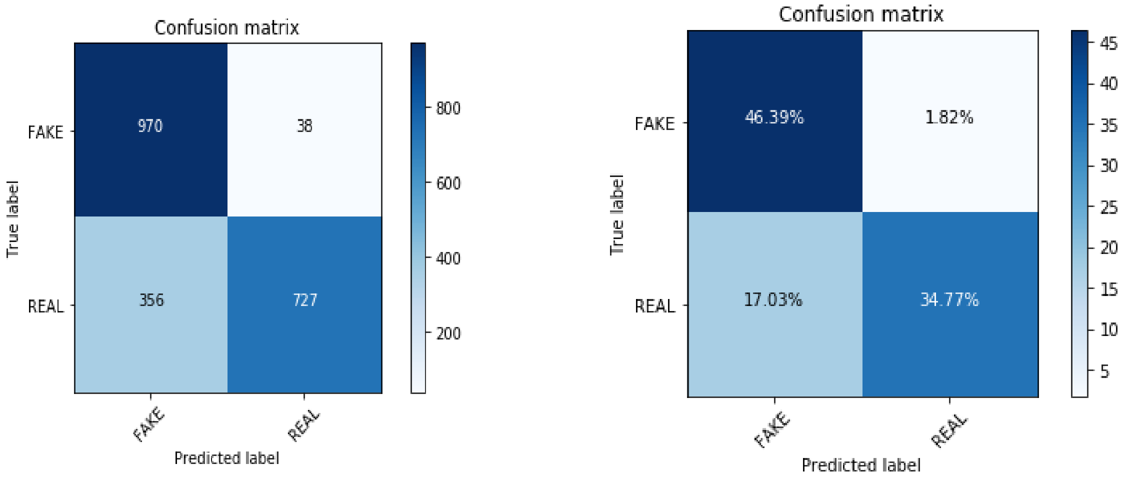


(a) Value

(b) Percentile

Figure 4.7: Confusion Matrix of SVM(no parameter tuning) on evaluation dataset

Then we tuned the value of the parameter C, which is the penalty parameter for error term with default value of 1.0. We set it's value to 1000.0 and now the SVM was working better giving us the accuracy of 81.2%. Below is given the confusion matrix.



(a) Value

(b) Percentile

Figure 4.10: Confusion Matrix of SVM(probability = True , C = 1000.0) on evaluation dataset

We could not get a better result using SVM and the building process of this model is very time

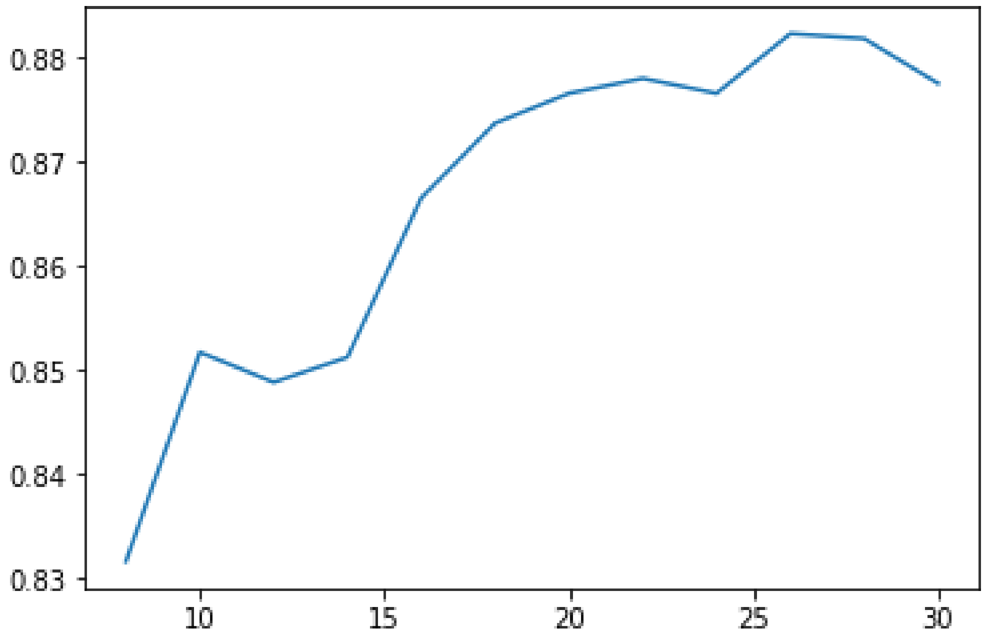
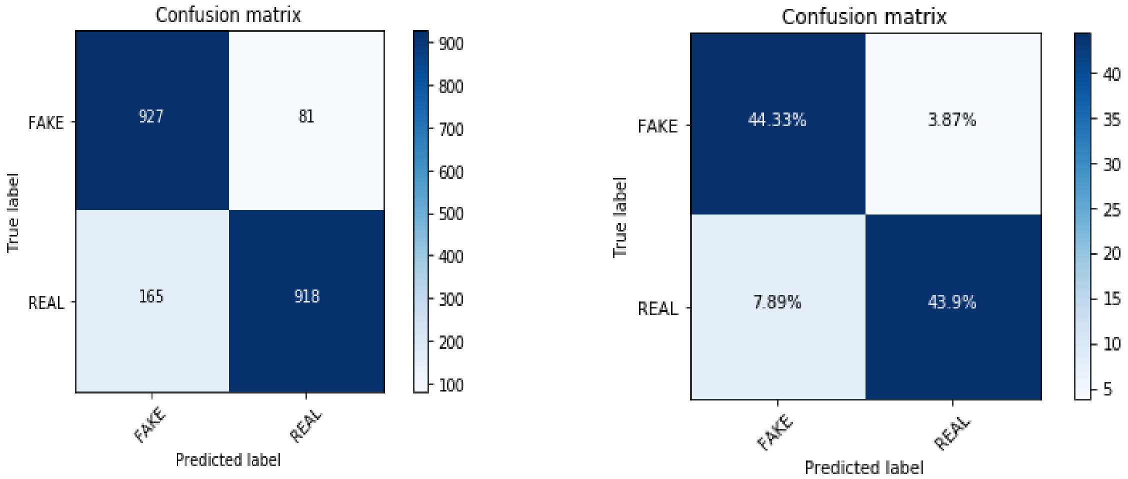


Figure 4.11: Different accuracy rates against different number of trees

Using Random Forest Classifier we got our highest accuracy rate of 88.2%, when we set number of trees to 26. Below is the confusion tree shown for this task.



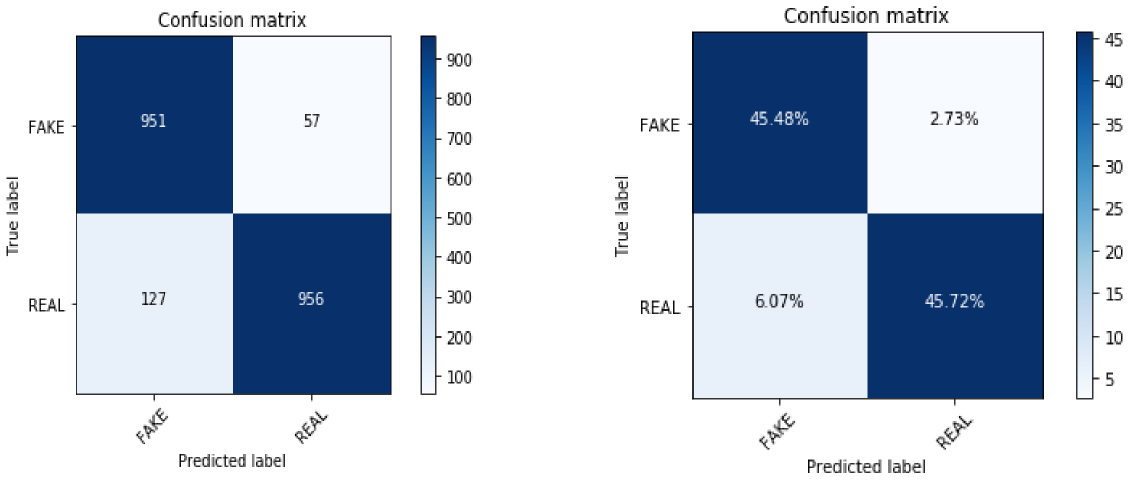
(a) Value

(b) Percentile

Figure 4.14: Confusion Matrix of Random Forest Classifier(n_estimators = 26 , criterion = ’entropy’) on evaluation dataset

4.4.4 Passive Aggressive Classifier

We actually acquired our highest accuracy rate Passive Aggressive Classifier. After some parameter tuning we got the accuracy rate of 91.2% after setting number of maximum iteration to 50 and random state to 17. Below is given the confusion matrix.



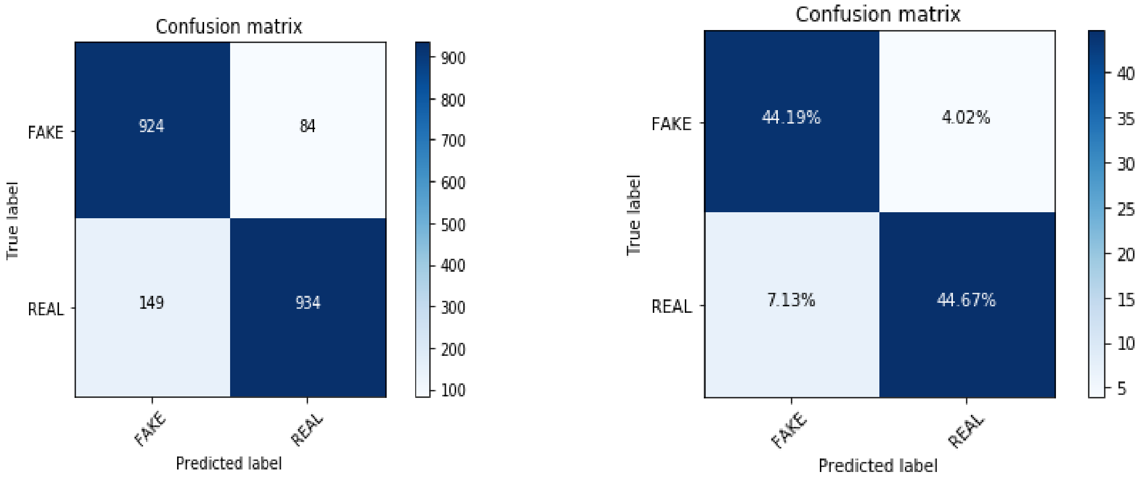
(a) Value

(b) Percentile

Figure 4.17: Confusion Matrix of Passive Aggressive Classifier(max_iter=50 , random_state = 17) on evaluation dataset.

4.4.5 XGBoost

XGBoost is combination of gradient boosted decision trees. This algorithm is very fast and provides high performance. We got a fair accuracy rate of 81.2% using naive bayes classifier and 88.2% using Random Forest Classifier. So, now we tried XGBoost in our dataset. Using XGBoost we got the accuracy rate of 88.9%. The confusion matrix is given below.



(a) Value

(b) Percentile

Figure 4.20: Confusion Matrix of XGBoost model on evaluation dataset

Accuracy Graph -

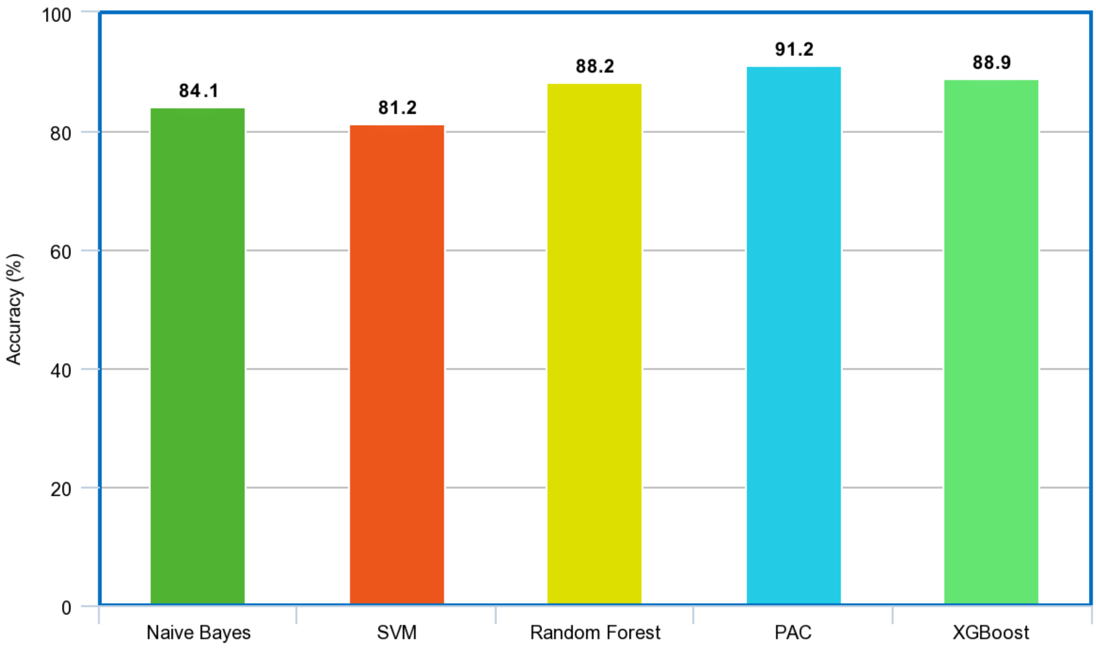
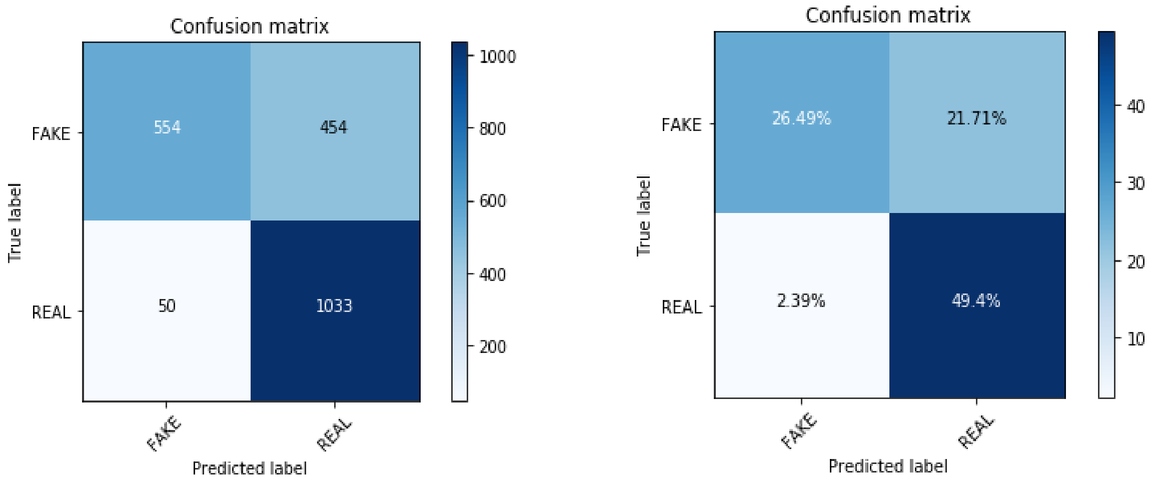


Figure 4.21: Accuracies for different classifiers

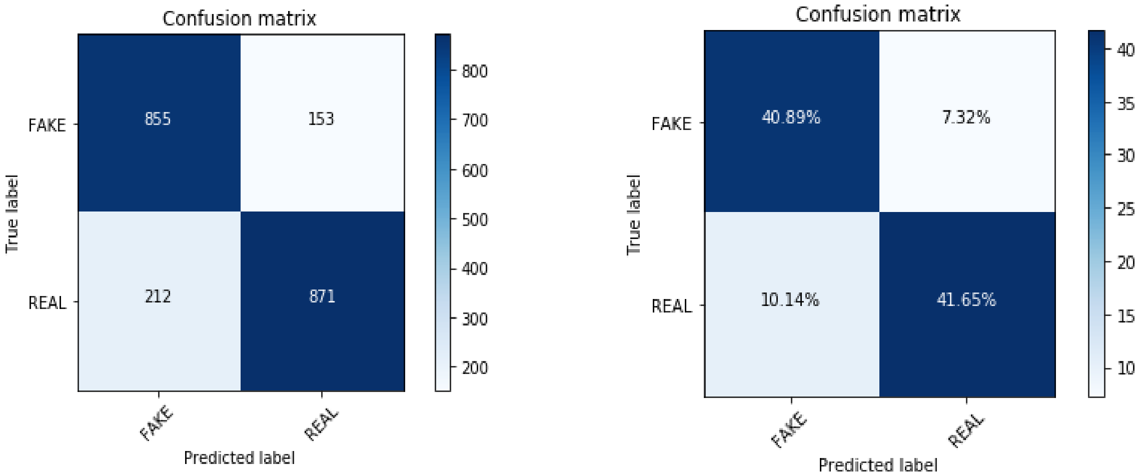


(a) Value

(b) Percentile

Figure 4.24: Confusion Matrix of Passive Aggressive Classifier for Word2Vec approach on evaluation dataset.

The XGBoost worked a bit better, giving us the accuracy rate of 82.5%. Below is the confusion matrix.



(a) Value

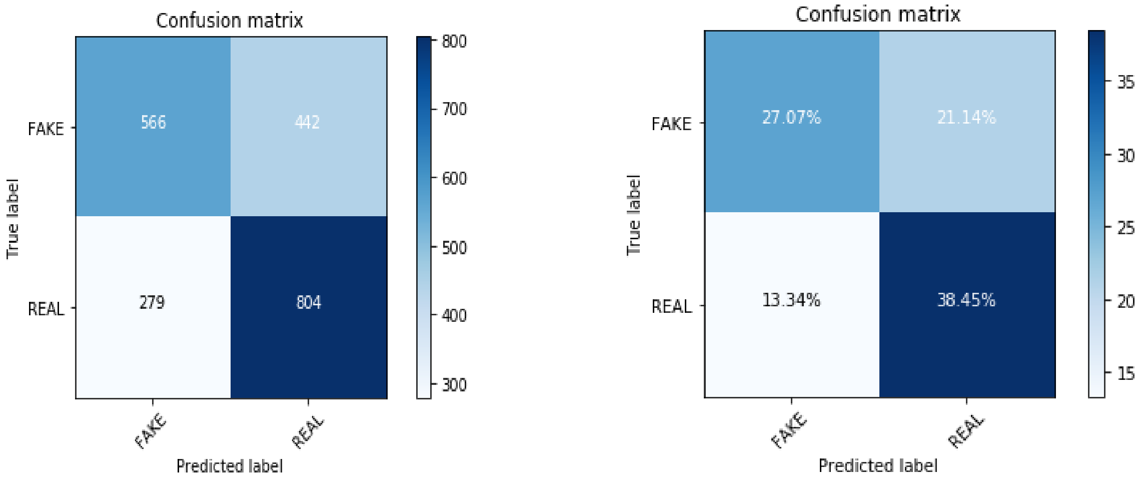
(b) Percentile

Figure 4.27: Confusion Matrix of XGBoost model for Word2Vec approach on evaluation dataset.

We thought Word2Vec mean multiplied with TF-IDF would give us a better result, but unfortunately it didn't.

4.8 Doc2Vec approach

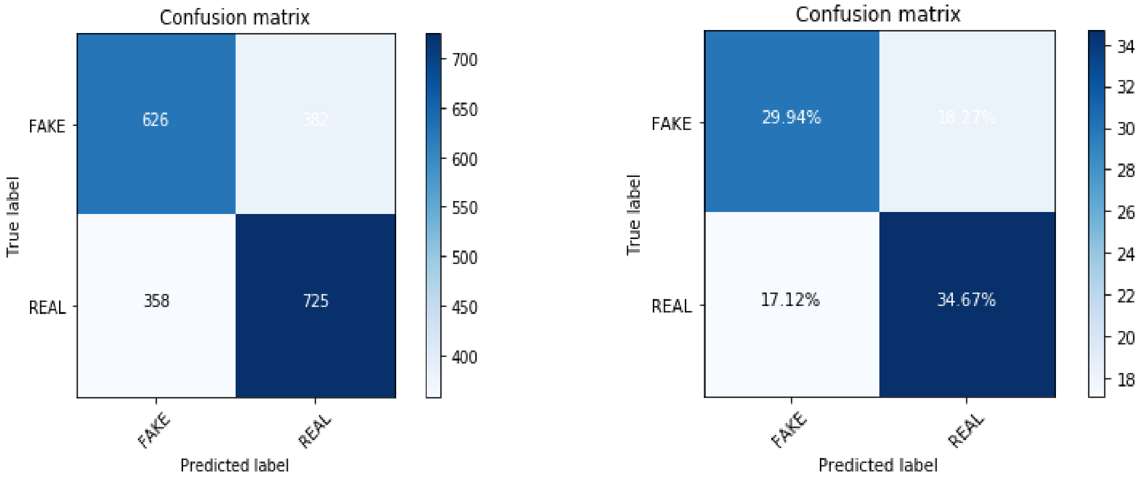
Now we tried our luck with the Doc2Vec model of *gensim*. This Doc2Vec model takes a article or sentence or document and returns a vector. This would be ideal for us. But using this approach we could not get a satisfactory result. Using Doc2Vec vector as a feature Passive Aggressive model would give us a F1-score of 65.5% and XGBoost would give us an F1-score of 64.6% on evaluation dataset. Below is given the confusion matrices.



(a) Value

(b) Percentile

Figure 4.30: Confusion Matrix of Passive Aggressive Classifier for Doc2Vec approach on evaluation dataset.



(a) Value

(b) Percentile

Figure 4.33: Confusion Matrix of XGBoost model for Doc2Vec approach on evaluation dataset.

5.4 At a Glance

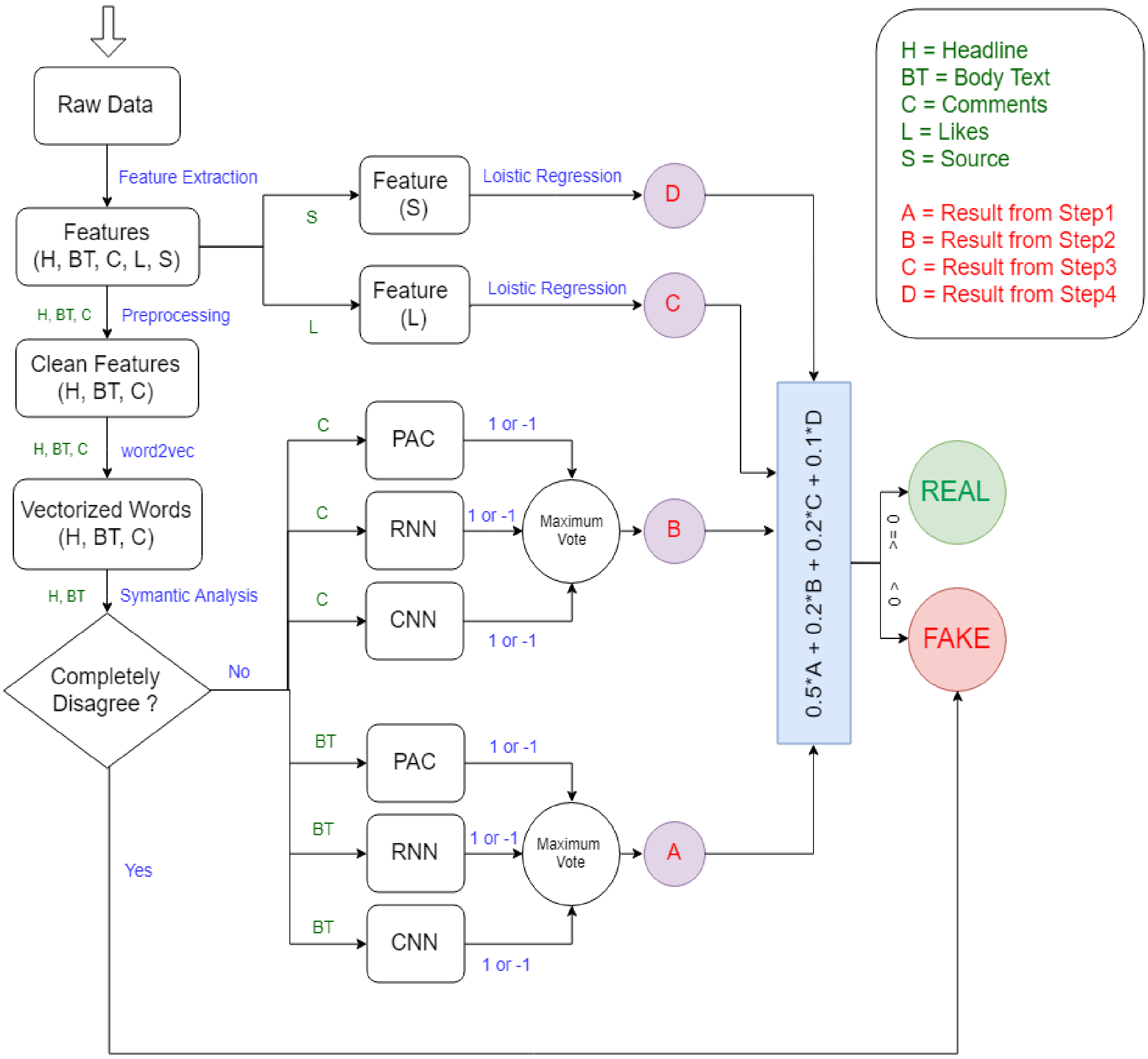


Figure 5.1: Work-flow of the Proposed Methodology