Detecting Real Or Fake Tweets

Era Sharma, MT19121
M-Tech CSE
IIIT DELHI
era19121@iiitd

Megha Mathur, MT19104
M-Tech CSE
IIIT DELHI
megha19104@iiitd

Saumya Jain, MT19098
M-Tech CSE
IIIT DELHI
saumya19098@iiitd

1 Problem Definition

The Problem Statement is about predicting the nature of Tweets i.e. which Tweets are talking about real disasters and which are talking about fake disasters.There were about 10,000 tweets in total which are classified into fake or real one. Task is to train different models according to these 10000 train data rows and predict the label of test data. Label 1 indicates Tweets are about real disasters and 0 for fake disasters.

2 Problem Background

Due to an increasing trend of need of information among the population and increasing dependence upon various social platforms like Facebook, Twitter, etc for information retrieval, there exists a need to detect whether such mediums are talking about real or fake news or disasters. Twitter has become a huge platform for communication in the time of Emergencies or disasters.But it was observed that sometimes the tweets or comments that were made on twitter are fake or unrealistic.So there exist a need to predict whether they are talking about real disaster or fake one. This prediction might help disaster relief agencies and other organisations to take decisions according to correct and current situation. To discriminate between real disaster tweets and fake one, Natural language processing(NLP) plays an important role. Various techniques like tf-idf,knn,similarity measures like resnik,lsa,n-gram etc provide aid in predicting such real or fake tweets as described below in subsequent sections.

3 Dataset

Dataset is taken from Kaggle named ”Real or Not? NLP with Disaster Tweets”. It’s motive is to predict which tweets are about real disasters and which are not. It comprises of 10,000 tweets which are classified into fake or real one. Task is to Each data point in test and training data set has information such as tweets text, keyword and the location from where the text has been sent from. Target column comprises of 2 values-0 or 1. 1 is for real disaster tweet and 0 for fake tweets.

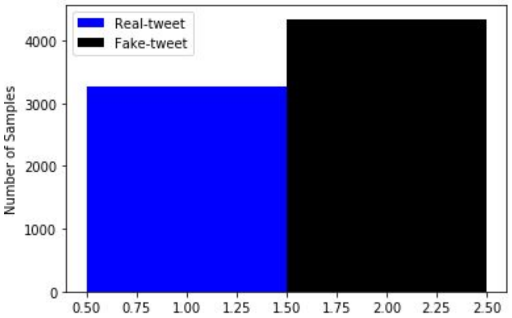


Figure 1: ratio of real and fake tweets

4 Literature Review

The research field is basically related to natural language processing because the data involves around the tweet text which are written in natural language and the second thing is classification because our ultimate goal is to classify the tweet as disastrous or not. The problem is oriented around tweets.Tweets are the text posted or written on online application called twitter which also acts as a source of information sharing.The problem discussed here is different from other problems like fake or real news,fake or real images and validity of rumours.But all these problems tries to achieve same objective that is classifying realness and fakeness on the basis of features and data.The text feature can be supported by other detailed features of text itself like text specific or tweet metadata.

The earliest mention of this problem comes in work done by Castillo et al.[1] They classified tweets based on the set of tweets related to a particular event.They used many features like twitter based,text based,user based,propagation based, but it lacked tweet level generalization and classification. Another work include that of Vosoughi et al.[3] classified rumours using propagation,text,user based features. Gupta et al.[2] classified tweets with associating images but again it was not generalized beyond test and training data.Also images can be subjected to cropping,editing which can itself question whether real or not. Volkova et al.[4] classified tweet as suspicious or trusted based on user account.But it doesnt gave much accurated results. The papers above used classifiers such as KNN,decision tree,Bagging,naive bayes for classification and TF-IDF,cosine,n-gram for feature extraction.

A survey of text similarity approaches[10] helped in analyzing the semantic and lexical properties of text. For classifing text, its meaning(semantics) and word arrangement(lexical) can play a major role. Similarity between text can be measured by analyzing which words are correlated or occurs together.The similarity measures discussed here is Latent Semantic analysis :which finds hidden topics in a corpus and semantically similar word under that topics,Pointwise Mutual Information techniques:finds co-occuring words in a text using syntax probability,N-gram based similarity which is a syntax based similarity technique finding common n words among 2 sentences, concept similarity measure and WordNet which finds similarity between two concepts . After reading and analyzing related work we can proceed with our working.

5 Proposed Solution

To tackle the problem of classifying the tweets into real and fake one, we have implemented various NLP models such as tf-idf, Naïve Bayes, KNN, Decision-Tree classifer, etc. Initially training dataset is divided into train data and validation data and then various NLP models as mentioned above will be applied to train the given dataset. Accuracy score of each model is calculated and on the basis of this score,best model to handle this problem and selected and then it is applied to predict the target(i.e whether real or not) column of the test dataset.

6 Baselines Created

We have taken tf-idf cosine similarity as our baseline model.Various preprocessing has been done before applying models such as tf-idf based technique, naive bayes,knn,decision tree,etc as described in section below. We have trained and tested our data on the basis of different models and calculated accuracy of each models.

6.1 Preprocessing

Various preprocessing steps has been done to make the available dataset efficient:-

- Text column is tokenised using nltk.word.tokenize.
- Text normlisation is done by converting each text into lower case.
- Then stop words are removed from the text.
- Stemming of words is done using porter stemmer.
- Lemmatization is then performed using wordnet lemmatizer.
- We have replaced NaN value in location with 'unknown' and in keyword with 'none'

6.2 tf-idf based cosine similarity

tf-idf technique describe the importance of word to a document in the corpus or collection. We have applied an algorithm to classify given tweets in two categories i.e real and not real. We have made a vector of terms of each class and then another vector for tweet which we are categorising is made. Cosine similarity is calculated between vector of that tweet with vector of both the classes.Tweet is assigned to the class having higher cosine similarity. This model shows the accuracy of 69.86 percent.

6.3 Features Extraction

Feature Extraction is the process of retrieving useful features or attributes from the given dataset which could facilitates the process of training data accurately and predicting the test labels efficiently. In the provided dataset, there are only three features that are being provided and they are :- Keywords, location and Text itself. Among these, some of location and keyword values are found to be empty. Therefore there exists a need to extract new features, so that we could apply multi-dimensional models accurately. Some of the features extracted are as follows:-

- Count of exclamation mark(!)
- Count of Question mark(?)
- Count of hash tag(#)
- length in characters
- length in word
- Count of URL
- Count of @
- Count of uppercase
- Count of lowercase

After selecting these features, we have created features using similarity scores. Resnik similarity score (described in next section) of each text is calculated with respect to real and fake data and reported as features respectively.Similarly LSA similarity score is taken into consideration.

7 Other Proposed Models

7.1 Naive Bayes

Naive-Bayes is a probabilistic classifier, which takes Bayes theorem into consideration.In Naive Bayes approach, we have made a dictionary for unigrams and their term frequency which are appearing in both the classes. By using term frequency dictionary, we have calculated a score. On the basis of this score,we classify the given tweet in real or non real classes. The score is measured using the formula:-

Score=E(log((1+No. of times term t appear)/(total terms in class c+ total terms in vocabulary))

It is measured for all the terms appeared in the given tweet.Accuracy obtained is 0.7284.

7.2 Decision Tree

Being a good model for classification We have selected Decision Tree as our third model. As in all models here also we have preprocessed and normalized tweet text from training data.After preprocessing document term tfidf matrix is created and the matrix is divided into training and testing data with test size 0.2. The training data along with target column (80 percent)is fitted over decision tree model and the model is then made to predict the target values of another 20 percent data the accuracy for this model is 0.7314510.

7.3 K Nearest Neighbour

The fourth model is K Nearest Neighbour, here we have taken the value of K as 6.The data is preprocessed and a dictionary is made, which contains all the words in the text of the data with its frequency.Vector is made using CountVectorization of the unique tokens.Validation set is made. Data is divided into training data and testing data with test size of 0.2. Model is fitted over the training data and prediction values of test data is done with the accuracy of 0.69.

8 Similarity Measures

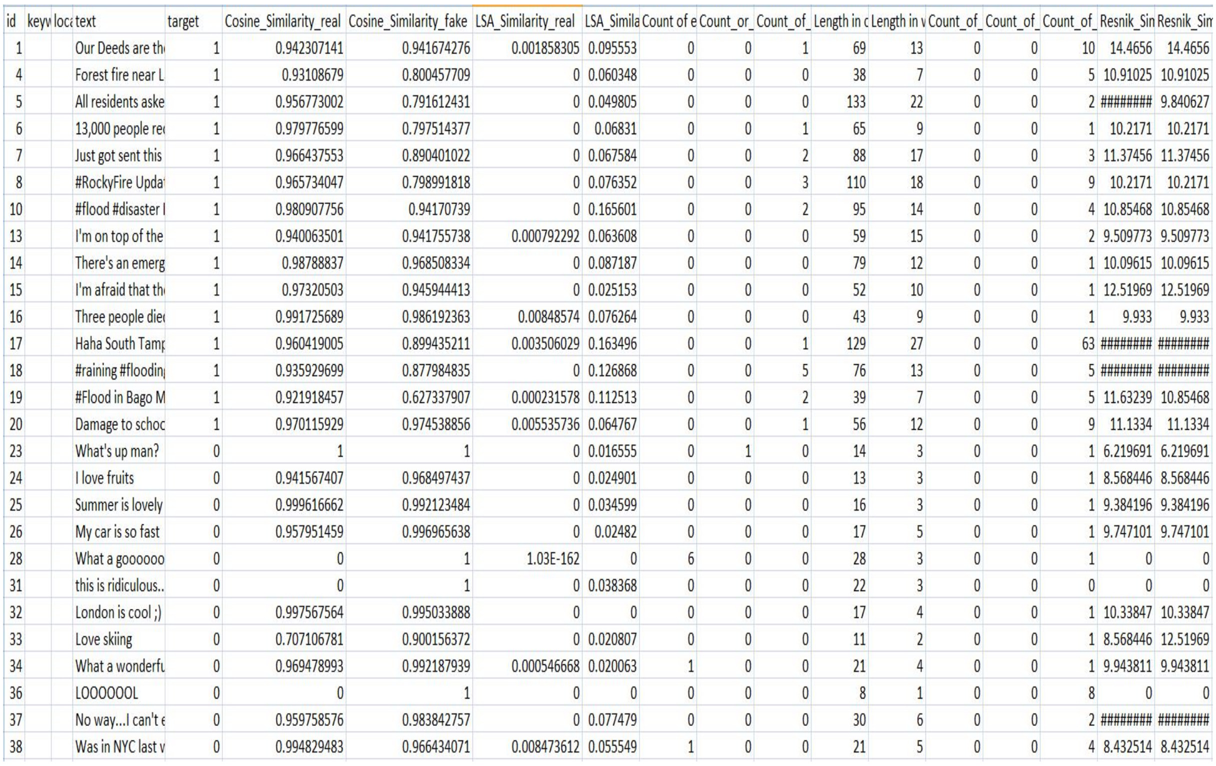
Similarity Measures are meant to measure the closeness of two thing or word either structurally or semantically. If two words are more alike in terms of lexical structure or meaning then they have high similarity score and vice-versa. In classification problems, Semantic measures are required to predict the closeness of two texts or words so that they can be properly classified. We have explored and implemented two types of similarity measures in our dataset and they are as follow:-

8.1 Semantic Similarity

Semantic similarity is knowledge based similarity that finds out the degree with which two words are similar according to the semantic-network. NLTK i.e Natural language toolkit is the most popular semantic-network to measure such similarities between the words.Words that are similar in context of their meaning is assigned similarity score of 1 and similarity measure varies from 0-1.We have used Resnik and LSA semantic similarity algorithms to classify our data into real and fake one.

8.1.1 Resnik Similarity

It is a Knowledge based similarity measure which calculates similarity between two words on the basis of semantic or information content. Resnik similarity score is calculated using the Information content. Information Content is computed for least common subsumer, it is the count(frequency) of word or concept found in the text corpus.Resnik



id	keyw	loca	text	target	Cosine_Similarity_real	Cosine_Similarity_fake	LSA_Similarity_real	LSA_Simila	Count of e	Count_or_	Count_of_	Length in c	Length in v	Count_of_	Count_of_	Count_of_	Resnik_Sin	Resnik_Sim
1			Our Deeds are th	1	0.942307141	0.941674276	0.001858305	0.095553	0	0	1	69	13	0	0	10	14.4656	14.4656
4			Forest fire near L	1	0.93108679	0.800457709	0	0.060348	0	0	0	38	7	0	0	5	10.91025	10.91025
5			All residents aske	1	0.956773002	0.791612431	0	0.049805	0	0	0	133	22	0	0	2	########	9.840627
6			13,000 people re	1	0.979776599	0.797514377	0	0.06831	0	0	1	65	9	0	0	1	10.2171	10.2171
7			Just got sent this	1	0.966437553	0.890401022	0	0.067584	0	0	2	88	17	0	0	3	11.37456	11.37456
8			#RockyFire Upda	1	0.965734047	0.798991818	0	0.076352	0	0	3	110	18	0	0	9	10.2171	10.2171
10			#flood #disaster	1	0.980907756	0.94170739	0	0.165601	0	0	2	95	14	0	0	4	10.85468	10.85468
13			I'm on top of the	1	0.940063501	0.941755738	0.000792292	0.063608	0	0	0	59	15	0	0	2	9.509773	9.509773
14			There's an emerg	1	0.98788837	0.968508334	0	0.087187	0	0	0	79	12	0	0	1	10.09615	10.09615
15			I'm afraid that th	1	0.97320503	0.945944413	0	0.025153	0	0	0	52	10	0	0	1	12.51969	12.51969
16			Three people die	1	0.991725689	0.986192363	0.00848574	0.076264	0	0	0	43	9	0	0	1	9.933	9.933
17			Haha South Tam	1	0.960419005	0.899435211	0.003506029	0.163496	0	0	1	129	27	0	0	63	########	########
18			#raining #floodin	1	0.935929699	0.877984835	0	0.126868	0	0	5	76	13	0	0	5	########	########
19			#Flood in Bago M	1	0.921918457	0.627337907	0.000231578	0.112513	0	0	2	39	7	0	0	5	11.63239	10.85468
20			Damage to schoo	1	0.970115929	0.974538856	0.005535736	0.064767	0	0	1	56	12	0	0	9	11.1334	11.1334
23			What's up man?	0	1	1	0	0.016555	0	1	0	14	3	0	0	1	6.219691	6.219691
24			I love fruits	0	0.941567407	0.968497437	0	0.024901	0	0	0	13	3	0	0	1	8.568446	8.568446
25			Summer is lovely	0	0.999616662	0.992123484	0	0.034599	0	0	0	16	3	0	0	1	9.384196	9.384196
26			My car is so fast	0	0.957951459	0.996965638	0	0.02482	0	0	0	17	5	0	0	1	9.747101	9.747101
28			What a gooooo	0	0	1	1.03E-162	0	6	0	0	28	3	0	0	1	0	0
31			this is ridiculous..	0	0	1	0	0.038368	0	0	0	22	3	0	0	0	0	0
32			London is cool ;)	0	0.997567564	0.995033888	0	0	0	0	0	17	4	0	0	1	10.33847	10.33847
33			Love skiing	0	0.707106781	0.900156372	0	0.020807	0	0	0	11	2	0	0	1	8.568446	12.51969
34			What a wonderfu	0	0.969478993	0.992187939	0.000546668	0.020063	1	0	0	21	4	0	0	1	9.943811	9.943811
36			LOOOOOOL	0	0	1	0	0	0	0	0	8	1	0	0	8	0	0
37			No way...I can't	0	0.959758576	0.983842757	0	0.077479	0	0	0	30	6	0	0	2	########	########
38			Was in NYC last	0	0.994829483	0.966434071	0.008473612	0.055549	1	0	0	21	5	0	0	4	8.432514	8.432514

Figure 2: Features Extracted

similarity basically works on the Word-net nouns. Formula used is as follow:-

$$SimilarityScore = -log(P(c)) * (LCS) \quad (1)$$

Here, P(c) is the probability that concept c's instance lies in the corpus.The accuracy of resnik similarity is 0.5538.

8.1.2 Latent Semantic Analysis

Latent Semantic Analysis(LSA) is a Topic Modelling Technique(one of the text mining technique) used to find topics in a collection of documents by grouping semantically similar words in a topic and associate each document with one of these topics based on similarity score between words in document and in topic.

LSA works by taking input a document-term matrix,number of topics(classes) and number of words in each topic to be returned.The output is topics and number of words in each topic. we implemented LSA here by creating 2 tfidf matrices, one with documents having target 1(real disaster) and second having document with target 0(fake) to have sematically similar words in each topic.Then we used a library function SVD(singular vector division) to decompose above 2 matrices into 3 matrices (1:document-topic,2:topic-topic,3:topic-term their multiplication gives the semantic score).Thus with input as first matrix,1000 word,2 topics we got 2 topics of semantically similar words then by manual identification we selected 1 topic with more number of disastrous words.similarly with second matrix,1000 words,2 topics we selected one of the topic as non disastrous.Then classified a document as real or fake by :-
score=E(similarity score of words matched in a topics)
A topic with higher score is the class assigned to that document.The accuracy of this model is 0.572.

8.2 Lexical Similarity

It is a similarity measure that helps to find out the degree to which two word sets are similar. It is also called word level similarity. It helps to expand the scope of the words lexically to improve similarity.

8.2.1 N gram

N gram is the sequence of n consecutive words. This model is used to estimate the probability of a word, given conditional probability of n-1 previous words. Bigram is used in this project. In bigram the value of n is 2, It is used to estimate the probability of a word given conditional probability of the previous word. Laplace smoothing is applied. The formula used is as follows:

P add-1 (W i—W i-1) = (count(W i-1, W i) + 1) / count(W i-1) + V.

Here, P add-1 is the probability of bigram, V is the vocab size. Data is divided into training data and testing data with test size of 0.2. For training, data is divided further into 2 parts on the basis of label(Real or fake) and probabilities are calculated for both the parts. In testing, bigram probabilities are calculated for both the parts. The label is assigned to the part which has the highest probability. The accuracy of this model is 0.5456.

8.3 Linear Regression

Linear Regression model is used to find the relationship between the dependent and independent variables. A line is obtained from the training data, in which the training data best fits.The distance from the line and the data points are calculated. This distance is called error. Different parameters are used to minimize this error. basically reduce the distance between calculated and real targets. we calculate y' calculated values of target) using formula of line.

$$y' = a + q1 * x1 + q2 * x2 + ... + qn * xn. \quad (2)$$

where a is intercept which minimizes the distance between real and calculated targets. n is the number of samples q is regression coefficients whose values are improved such that it minimize the error or cost. x is particular sample values

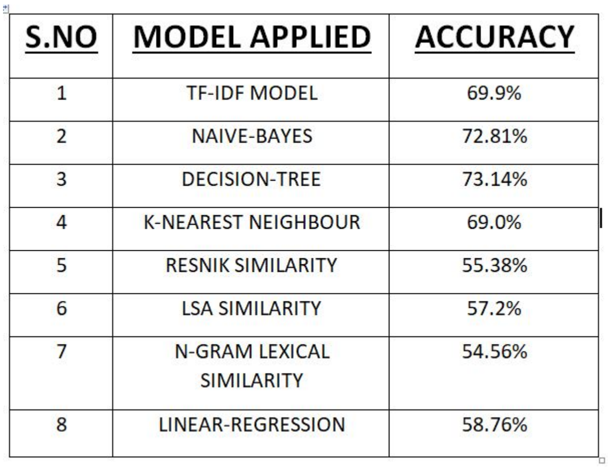
first we normalized the tfidf values for each column and sample value using mean and standard deviation

$$x = (x - u)/sigma \quad (3)$$

Here u= mean of a column and sigma =standard deviation. Then we trained the model by by calculating the coefficients using hypothesis function this hypothesis is calculated by multiplying 'n' coefficients with 'n' sample values.hypothesis function helps in calculating the y' values. we use Batch Gradient Descent(BGD) function which is used to optimize the coefficients to minimize the cost and calculate the cost/error.This BGD runs for a number of iterations(3000 here) which is needed to optimize the coefficients and cost.a learning rate of 0.0001 is used to increase the learning of BGD function for getting optimum coefficients. After a number of iterations when we get cost and efficient coefficients we then predicted the value of our test.csv matrix by calling the hypothesis function and inside it BGD function.The accuracy of this model is 0.5876.

9 Experimental Result

Figure 3 shows different models along with their accuracy score. It was observed and concluded that similarity measures like resnik,LSA and n-Gram are less reliable for the dataset provided, whereas Naive-Bayes and decision-tree is providing the best accuracy among all the models applied. High-dimensional model like linear regression is not giving accurate result due to less availability of good features.



S.NO	MODEL APPLIED	ACCURACY
1	TF-IDF MODEL	69.9%
2	NAIVE-BAYES	72.81%
3	DECISION-TREE	73.14%
4	K-NEAREST NEIGHBOUR	69.0%
5	RESNIK SIMILARITY	55.38%
6	LSA SIMILARITY	57.2%
7	N-GRAM LEXICAL SIMILARITY	54.56%
8	LINEAR-REGRESSION	58.76%

Figure 3: Model vs Accuracy Obtained

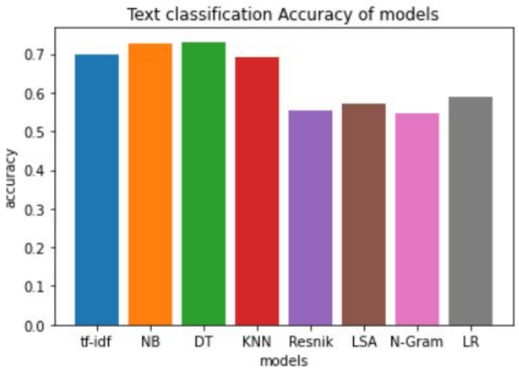


Figure 4: Graph showing models with their accuracy

10 Analysis

The documents are preprocessed by stemming and tokenizing text in the train document and a tfidf based cosine similarity is created as a baseline model. Various machine learning models are used such as naive bayes, KNN and Decision Tree. Decision tree worked well as per the accuracy obtained. As we are predicting whether the tweets is real or fake, we tried to use various properties of text such as semantic and lexical property. So we implemented some semantic similarity measures such as Latent Semantic Similarity, Resnik similarity and and lexical similarity measures such as N gram. Experimentally, LSA worked better on the basis of accuracy but still accuracy was not good enough as there is less difference in textual content of both the classes i.e many words were common in both the classes. Hence, we turned towards feature extraction technique. We extracted features such as similarity scores of both classes of semantic and lexical models and other features such as count of ”#”,”!”,”?”,”@”,Upper case letters,lower case letters,URL,length of words,length of sentence. Then we applied linear regression model (one of the high dimensional model) using the extracted features.

11 Conclusion

The problem statement was to predict the nature of tweets about disaster, whether it’s real or fake. We applied different models including tfidf based cosine similarity model, Naive Bayes, Decision Tree, K-Nearest Neighbour, Resnik Similarity, LSA, N-gram and Linear regression. The accuracy of Decision Tree is the highest among all the models applied. Therefore, we are able to predict the labels of test data with an accuracy of nearly 73%.
For future work this classification can be extended to classify texts other than tweets.Also including real time text classification can be the most effective application of this classification.As the training set was with limited features more sophisticated feature can be extracted by finding or storing more information about training set.

12 References

1. Castillo C, Mendoza M, Poblete B (2011) ,”Information credibility on twitter”. In: Proceedings of the 20th international conference on World wide web, ACM, pp 675-684

2. Gupta A, Lamba H, Kumaraguru P, Joshi A (2013) Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: Proceedings of the 22nd international conference on World Wide Web companion, pp 729-736

3. Vosoughi S, Mohsenvand M, Roy D (2017),” Rumor gauge: Predicting the veracity of rumors on twitter.”, ACM Transactions on Knowledge Discovery from Data 11:1-36

4. Volkova S, Shaer K, Jang JY, Hodas N (2017) ”Separating facts from Linguistic models to classify suspicious and trusted news posts on twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics”, vol 2, pp 647-653

5. Rada Mihalcea,Courtney Corley,Carlo Strapparava (2006),”Corpus-based and Knowledge-based Measures of Text Semantic Similarity”,AAAI’06: Proceedings of the 21st national conference on Artificial intelligence - Volume 1July 2006 Pages 775–780.

6. Samuel Fernando and Mark Stevenson,Department of Computer Science,University of ,Sheffield,S1 4DP, UK,”A Semantic Similarity Approach to Paraphrase Detection”.(2009)

7. Christina Boididou ,Symeon Papadopoulos ,Markos Zampoglou,Lazaros Apostolidis ,Olga Papadopoulou ,Yiannis Kompatsiaris,”Detection and Visualization of Misleading Content on Twitter”,International Journal of Multimedia Information Retrieval volume 7,pages71–86(2018).

8. Kushal Agarwalla, Shubham Nandan, Varun Anil Nair, D. Deva Hema,”Fake News Detection using Machine Learning and Natural Language Processing”,International Journal of Recent Technology and Engineering (IJRTE),ISSN: 2277-3878, Volume-7, Issue-6, March 2019.

9. Maarten S. Looijenga,University of Twente,”The Detection of Fake Messages using Machine Learning”,

10. Wael H. Gomaa,Aly A. Fahmy,”A Survey of Text Similarity Approaches”,International Journal of Computer Applications (0975 – 8887),Volume 68– No.13, April 2013