Fake News Detection In HealthCare Domain

Students, Third Year Computer Engineering Dept., VESIT, Chembur, Mumbai-71

Prof.Lifna C S
Dept. of Computer Engineering
VESIT
Mumbai, India
lifna.cs@ves.ac.in

Mr. Pratik Chainani
Dept. of Computer Engineering
VESIT
Mumbai, India
2018.pratik.chainani@ves.ac.in

Ms.Khushboo Bhatia
Dept. of Computer Engineering
VESIT
Mumbai, India
2018.khushboo.bhatia@ves.ac.in

Mr. Laveen Dawani
Dept. of Computer Engineering
VESIT
Mumbai, India
2018.laveen.dawani@ves.ac.in

Abstract:
In the past years social media platforms like twitter ,whatsapp, facebook have become a source of spreading information quickly and widely . With the easy internet accessibility , people are free to spread information that they believe . This fake information is dreadful especially if it's related to healthcare . Fake remedies , theories with no proper evidence about illness are spreaded all over the internet . In the current times of pandemic , all our phones were buzzing with made-up theories , remedies , diagnosis of the virus . Social media has seen a surge in spreading information . It is helpful but there is a need for verification of such information .
In this paper we will discuss the Fake News Detection model that will classify Covid-19 related tweets as Real or Fake. We have classified on the basis of language used in tweets along with judging the authenticity of links shared and the people tagged.

I.Introduction:

There is a spike in consumption of news from social media sites . But the quality of news can not be trusted as there is open network for everyone to spread self made theories, conspiracy theories, false information with or without intension of creating a hoax in people

.Sometimes there is unintentional spread of fake news for example someone proposed a self made logic behind connection of covid-19 vaccine and Menstrual cycle which somehow created a fear in general public. When such articles are being forwarded all over social media specially related to health then there is a risk . After referring to the methodology of various available websites like altnews.in , politifact , we got to know there are humans working to manually check facts.

Our vision is to create an automated fact checking model. We used a syntactic stylometric approach for deception detection and taking into account usernames and domain of urls used in the tweets. With a stylometric approach we are considering a specific domain i.e health care and more precisely covid-19 related news for fact checking.Based on linguistic approach , we used word embedding methods for detection fake news and with frequency based embedding we got higher accuracy of 90% . After ensembling authenticity of user names and domain names of url we were able to increase the accuracy by 92% . This paper shows our approach of detecting fake news based on Natural language processing

II.Literature Survey:

In [4]. Wang, William Yang, set up a benchmark dataset for fake news detection .'LIAR' is the name of a publicly available dataset, which is used for fake news detection.It includes 12,836 short statements labeled for truthfulness, subject, context/venue,speaker, state, party, and prior history. The LIAR dataset includes12.8K human labeled short statements from POLITIFACT.COM's API, and each statement is evaluated by a POLITIFACT.COM editor for its truthfulness.The dataset statistics are observed and studied.Neural Networks framework for integrating text and meta-data has been used for implementation.We can also label NEWS as "false","barely true", "half true", "mostly true".Compared to prior datasets,LIAR is an order of a magnitude larger, which enables the development of statistical and computational approaches to fake news detection.

CSI [5]. Is a Hybrid Deep Model for Fake News Detection.There are several characteristics that are generally agreed upon detection of fake news, relating to the text of an article, the response it receives, and its source.

1.type of text of that article has
2.Response it receives by the reader3.Source of that news
Solution model consist of three modules vizCSI-Capture Scorel Integrate-1.Capture-this module receives text and response on the the article and process with help of Recurrent Neural Network(RNN)2.Score-scores the user(one who promote fake news) on basis of its previous news and its authenticity 3.integrate-integrate both of

the above modules and give the final result

Some like it hoax[6] Luca de Alfaro have presented two classification techniques/alg
1.logistic regression
2.a novel adaptation of boolean crowdsourcing algorithm.Following are the contribution by above article i) the proposal of a novel way to identify hoaxes onSocial Networking Sites(SNSs) based on the users who interacted with them rather than their content ii) an improved version of the harmonic crowdsourcing method, suited to hoax detection in SNSs iii) the application on Facebook and, in particular, on a representative dataset obtained from the literature.They also analyzed the extent to which performance depends on the community of users naturally aggregating around pages of similar content. They showed that the harmonic BLC algorithm can transfer information across pages: even when only half of the pages are represented in the training set, the performance is above 99%
Both algorithms provide good performance, with the harmonic BLC algorithm providing accuracy above 99% even when trained over sets of posts consisting of 0.5% of the full dataset (or about 80 posts). This suggests that the algorithms can scale up to the size of entire social networks, while requiring only a modest amount of manual classification

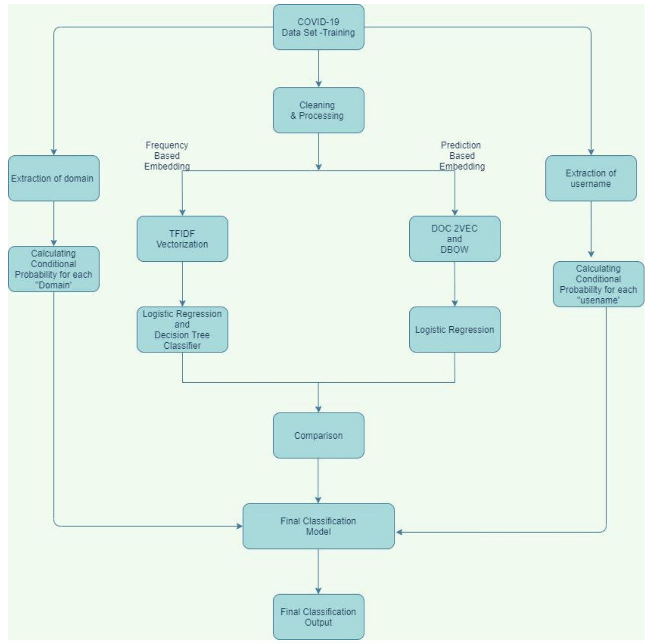
Pérez-Rosas, Verónica In [10] mention that The proliferation of misleading information in everyday access media outlets such as social media feeds, news blogs, and online newspapers have made it challenging to identify trustworthy news sources, thus increasing the need for computational tools able to provide insights into the reliability of online content. In this paper, they focus on the automatic identification of fake content in online news. Their contribution is twofold. First, they introduce two novel datasets for the task of fake news detection, covering seven different news domains. They describe the collection, annotation, and validation process in detail and present several exploratory analyses on the identification of linguistic differences in fake and legitimate news content. Second, they conduct a set of learning experiments to build accurate fake news detectors. In addition, they provide comparative analyses of the automatic and manual identification of fake news

III.Dataset:

The dataset [11] for CONSTRAINT COVID-19 Fake News Detection in English challenge was provided by the organizers on the competition website3 . It consists of data that have been collected from various social media and fact checking websites, and the veracity of each post has been verified manually. The “real” news items were collected from verified sources which give useful information about COVID-19, while the “fake” ones were collected from tweets, posts and articles which make speculations about COVID-19 that are verified to be false. The original dataset contains 10,700 social media news items, the vocabulary size (i.e., unique words) of which is 37,505 with 5141 words in common to both fake and real news. It is class-wise balanced with 52.34% of the samples consisting of real news, and 47.66% of fake samples

IV.System Design:

The system consists of various modules which work in unison to predict the given piece of information based on the language used . We took only the text of the information into consideration for prediction .Preprocessing module works by cleaning the data and crawling web pages given in tweets for additional information.The clean data is given to word embedding modules , which tokenize using tfidf and doc2vec algorithms and a domain specific model is trained containing words from our domain-healthcare-covid19. Then comes the classification algorithms, we have used Logistic regression and decision tree classifier for classification and on based on results we will select final module which will be use for future prediction and manual testing
The following diagram represents the architecture for our project.



V.Methodology:

Data Preparing:

1. Text Preprocessing:
   The text in the tweets are written with symbols, hashtags, emojis so our first step is to process the textual data and filter out tweets before feeding it to our model.
2. Tokenization:
   After Filtering out tweets from noisy data we then removed stop words using NLTK toolkit . Tokenization of a sentence is a very important part of NLP , it means breaking down sentences or paragraphs into phrases or words and term them as individual tokens. We used the Natural Language toolkit, a library in python, for doing this.

Vectorization :

Tokens formed cannot directly be fed into Machine Learning Algorithms , so the are converted into vectors using word embedding techniques
Word Embeddings: Machine learning algorithms require numbers as inputs to perform any sort of job, be it

classification, regression . And with the huge amount of data that we are dealing with in the text format, it is imperative to extract knowledge out of it .We have used two techniques to compare the results .

1. Frequency based Embedding -TFIDF:

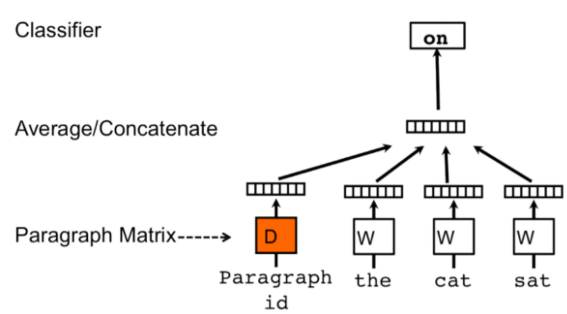
Doc1 Doc2

Term	Count	Term	Count
This is about Covid	1 1 2 4	This is about TFIDF	1 2 1 1

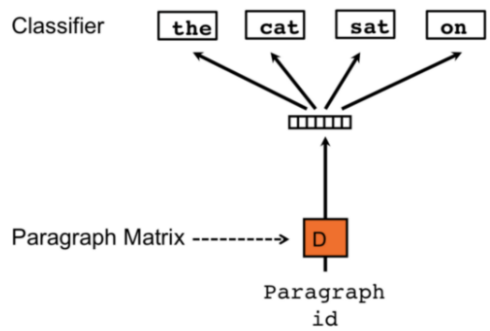
TF = (Number of times term t appears in a document)/(Number of terms in the document)
So, TF(This,Doc1) = 1/8
TF(This, Doc2)=1/5
IDF = log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.where N is the number of documents and n is the number of documents a term t has appeared in.
So, IDF(This) = log(2/2) = 0.
IDF(Covid) = log(2/1) = 0.301.
TF-IDF(This,Document1) = (1/8) * (0) = 0
TF-IDF(This, Document2) = (1/5) * (0) = 0
TF-IDF(Covid, Document1) = (4/8)*0.301 = 0.15
As you can see for Doc1 , TF-IDF method heavily penalises the word ‘This’ but assigns greater weight to ‘Covid’. So, this may be understood as ‘Covid’ is an important word for Doc1 from the context of the entire corpus.

2. Prediction Based Embedding-Doc2vec:
   Doc2vec uses an unsupervised learning approach to better understand documents as a whole.To implement this we are using the Gensim module .Doc2vec was created by Mikilov and Le in 2014. Mikolov was also one of the authors of the original word2vec research, which is another indicator that doc2vec is building on the word2vec architecture.(original work:https://arxiv.org/abs/1405.4053 Doc2vec is a extended model of word2vec .word2vec representation of 2 algorithms: Continuous Bag-of-Words model (CBOW) and the Skip-Gram model.

CBOW works to predict the probability of a word given a context. A context may be a single word or a group of words.Similarly Doc2vec consists of 2 architecture: namely Distributed Memory Model of Paragraph Vectors (PV-DM) and Distributed Bag-of-Words version of Paragraph Vector (PV-DBOW).



The architecture of PV-DM model



The architecture of PV-DBOW

After Vectorization we need to feed these vectors into our machine learning algorithms in order to Classify them as real or fake

Classification:
We have use Logistic Regression for classification
Logistic Regression:
Logistic regression is a s method for analyzing a the outcome which is dependent on two or more variables.Dichotomous variable (in which there are only two possible outcomes) is used to measure outcome

logistic regression finds the best fitting model for describing the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest:

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

where p is the probability of presence of the characteristic of interest. The logit transformation is defined as the logged odds:

$$= \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

and

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

But for our final model we do not need classification but need the probability of being real or fake , we can obtain this by using *predict_proba(X)* method in sklearn library . For example :
We have two classes in output i.e Real and Fake
Model. predict() method predicts the output class but model.predict_proba will give the probability of being Real or Fake .Now we have probabilities of outcomes from two models

Average of probabilities of both models:
We have taken the average of probabilities of Real and Fake news of both our models and termed as final prediction probability vectors.
Suppose $P_r(x)$ and $P_f(x)$ are prediction probabilities of tweet x and $p_r1(x)$ and $p_f1(x)$ are prediction probabilities obtained from first model and $p_r2(x)$ and

$p_f2(x)$ are prediction probabilities obtained from second model then

$$Pr(x) = (pr1(x) + pr2(x))/2$$

$$Pf(x) = (pf1(x) + pf2(x))/2$$

Domains AND Usernames :

Idea behind using Domains and Usernames is that they contain reliable information about the tweet , the domain name of the url given in the tweet specifically tells about what the tweet is about and whether it is from a trusted site. So we incorporate these two in our model by calculating probability vectors of both classes corresponding to both of them. We referred paper [1] for this approach of calculating probabilities

Steps to calculate the probability

- Extract username and domain name from the tweets
- Calculate how many times it appears in total and how many times it appears in real as well as in fake news
- Calculate Conditional probability

  For Domain name D:
  So probability vector of tweet x as Real given it has Url containing domain D is given by:

  $$Pr(x/D) = n(R)/(n(R) + n(F))$$

  And probability vector of tweet x as Fake given it has Url containing domain D is given by:

  $$Pf(x/D) = n(F)/(n(R) + n(F))$$

  Where n(R) is the number of occurrences of domain name in the real tweet and n(F) is the number of occurrences of domain name in the Fake tweet
  Similarly,
  For Username U
  Probability vector of tweet x as Real given that it has Username U is given by

  $$Pr(x/U) = n(R)/(n(R) + n(F))$$

  Probability vector of tweet x as Fake given that it has Username U is given by

  $$Pf(x/U) = n(F)/(n(R) + n(F))$$

  Where n(R) is the number of occurrences of Username in the real tweet and n(F) is the number of occurrences Username in the Fake tweet

This gives us new vectors as features for our final classification model.

For Final Classification we use Logistic regression , our feature vectors i.e probabilities of both classes based on text in tweets and probabilities of real and fake Urls and usernames .For the tweets that does not contain url or domain or both then classification will be based purely on text.

VI.Results:

Firstly we experimented with the text of the tweets only, we used tokenization and word embedding techniques for vectorization and logistic regression for classification and obtained following results

Word embeddings	Accuracy
Tfidf	0.902
Doc2vec	0.638

Then we incorporated usernames and Urls extracted from tweets to get more reliable results , with this model we got accuracy of 0.921.

VIII. Conclusion:

We first compared word embedding techniques (frequency based and Prediction based) for classification of tweets based on text used in it and then added Domain name of Urls and Usernames in the tweets as new attributes in classification along with the text of tweets and concluded that our approach was domain specific . Tfidf and Doc2vec are very domain specific methods for classification .

IX.Future Scope:

As the complications of detecting fake news increases, more and more research and development are required to be done by those working in the field of detecting fake news. More sophisticated and efficient model can be built by considering multiple parameters. We would like to extend our research to unstructured data like photos and videos which are more dangerous forms of fake news. Analyzing non-linear data gives an opportunity to research deep learning algorithms as well. Natural Language Processing could also be used to understand the brief meaning of text and compare them with multiple reliable news websites.

References:

1. A Heuristic-driven Ensemble Framework for COVID-19 Fake News Detection},author={Das, Sourya Dipta and Basak, Ayan and Dutta, Saikat},journal={arXiv preprintarXiv:2101.03545},year={ 2021}
2. Dataset:https://github.com/diptamath/covid_fake_news/blob/main/data/Constraint_Train.csv
3. Shu, Kai, et al. "Fake news detection on social media: A data mining perspective." ACM SIGKDD explorations newsletter 19.1 (2017): 22-36
4. Conroy, Nadia K., Victoria L. Rubin, and Yimin Chen. "Automatic deception detection: Methods for finding fake news." Proceedings of the Association for Information Science and Technology 52.1 (2015): 1-4.
5. Wang, William Yang. "" liar, liar pants on fire": A new benchmark dataset for fake news detection." arXiv preprint arXiv:1705.00648 (2017).
6. Ruchansky, Natali, Sungyong Seo, and Yan Liu. "Csi: A hybrid deep model for fake news detection." Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017.
7. Tacchini, Eugenio, et al. "Some like it hoax: Automated fake news detection in social networks."

arXiv preprint arXiv:1704.07506 (2017).

8. Tschiatschek, Sebastian, et al. ``Fake news detection in social networks via crowd signals." Companion Proceedings of The Web Conference 2018.
   Wang, Yaqing, et al. "Eann: Event adversarial neural networks for multi-modal fake news detection." Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining. 2018.
9. Pérez-Rosas, Verónica, et al. "Automatic detection of fake news." arXiv preprint arXiv:1708.07104 (2017).
10. Long, Yunfei. "Fake news detection through multi-perspective speaker profiles." Association for Computational Linguistics, 2017