Fake News Detection Using Machine Learning

Lilapati Waikhom[a] and Rajat Subhra Goswami[b]

Department of Computer Science & Engineering, NIT, Arunachal Pradesh, India

ARTICLE INFO

Article history:
Received 16 February 19
Received in revised form 15 March19
Accepted 05 April 19

Keywords:
Ensemble,
Fake News,
Liar dataset,
Classification,
XGBoost

ABSTRACT

Nowadays most of the people prefer the internet to access news as it is easy and cheap, but that results in wide spreading of fake news very fast. Fake news is often written with an ulterior motive to gain financially, politically, etc. with most of the time having a catchy headline which attracts users or it may also be accidental. But it affects so much to the people. Fake news detection has become a challenging topic nowadays. In this work, we use the LIAR dataset which is collected from POLITIFACT.COM for fake news detection and it is publicly available for use, which provide links to source documents for each case. In all the previous works, the accuracies are all around 30 percent on this dataset. In this work, we use model ensemble techniques to have better accuracy in predicting fake news using the LIAR dataset. We have also tried to simplify the problem statement into binary classification and deployed the same ensemble techniques to have an even better realistic approach for accurate calculation.

Peer review under responsibility of International Conference on Advancements in Computing & Management.

1. **Introduction**

The Internet has become compulsory in our life. It is now very easy to access the Internet than it was before. There is no doubt that many young people prefer the internet to get their news rather than the newspaper, radio, etc.[1]. The Internet provides many opportunities for us, we can search for anything on the internet to clear our doubt and for research purpose also. Simply saying, we can't even imagine our life without the internet. As more people are connecting to the Internet, they get most of their information content through it. In a country like India where Internet access has become cheap recently, a lot of people are accessing news through their digital devices. But when it comes to news publishing it creates so many issues. If it is about the news, the internet plays a very important role because through the internet, the news widespread very fast. There are so many consequences of fake news, it can cause harm to innocent people. Fake news may be made intentionally or accidentally to give harms to an individual or a group for any purposes, such as for political issues, for religious purposes and so on.

There have been many incidences of people getting hurt or getting killed because of rumors on the Internet. The creation of fake news generally increases during the time of the election in a country. The BBC news broadcaster has done research on Indian general election during 2014. The researchers [2] viewed about 16000 and 3000 accounts and pages from Twitter and Facebook respectively to learnhow fake news getspolarized in India. This research indicates a "strong and coherent" proliferation of right-wing ideology, while networks which promote left-wing ideology based fake news were found to be less organized and effective. Another research [3]by the BBC resulted that nearly 72% of Indian citizens are not able to differentiate between real facts from made-up ones. Altogether, these concludes that we need to expose people in India to the digital literacy to overcome the consequences of fake news in the country. In this paper, we first review the work done up to now, then we present the POLITIFACT dataset [4] and explain the feature extraction done on the dataset. After this, we present ensemble of multiple models and reclassify the problem as a binary classification rather than a multi-classification problem as presented by the POLITIFACT dataset. Then the evaluation and results are presented at the end of this paper.

2. **Related Work**

Automatic fake news detection has already been studied for some years. Rubin, et.al in [5] gave a hybrid approach which combines the linguistic features of a language with the network analysis approach. This method always may not suitable as the network information may be restricted or not available. In [6] as discussed by Rubin, et. al. has analyzed rhetorical structures and the relation between the various other structures of fake and truthful news sample from NPR's "Bluff the Listener". They have applied clustering to achieve 63% accuracy. In [7], Mihalcea and Strapparava showed that by using deep learning it is possible to differentiate between false and true information to some degree.

Older studies are mainly focused on lexical patterns of the language, Fend, et al. in [8] applied syntactic stylometry to text, which made it possible classify deceptive text by finding statistical or syntactic patterns. Text analysis is the major resource for fake news detection because of the well-known methods to analyze text. This linguistic-based classification is done by Veronica Perez-Rosas, et. al [9] for fake new detection and suggest that linguistic features are a major factor in the detection of fake news than real news. These approaches are heavily inclined toward language-based analysis and are somewhat limited[10] to overcome this we have to combine other features related to the news. They have combined metadata from Google and incorporated that to boost the classification by 3 % in F1-score for 6-label classification problem.

3. **LIAR Dataset**

Most of the datasets available contain short statements as the language used for political information broadcasting on TV interviews, Facebook posts and tweets on Twitter are mostly of short length statements,that's why the detection of fake news is more challenging. In this work, we use a publicly available benchmark dataset (LIAR dataset) collected from the website POLITIFACT.COM that has detailed report and URL to eachsource document sample to make the development of techniques for fake news detection automatic. This labeled dataset consists of thirteen different features in 12.8k samples of data that is manually labeled into various categories about politics which is analysed by the editor of POLITIFACT.COM and categories according to its truthfulness. The dataset contains speaker of each statement as 'speaker', speaker's job as 'speakerjob', state information of the speaker as 'stateinfo', party affiliation as 'partyaffiliation', context of the statement as 'context', and subjects of the news 'subjects'. The statements are labelled in six categories –true, mostly-true,barely true,pants-on-fire, false, and half-true. The number of data samples for all categories falls under the range from 2,063 to 2,638 except for the category pants-on-fire which has 1050 data samples in the whole dataset. The dataset contains the statements from the year 2007 to 2016. The dataset contains speakers as a mix of republicans and democrats, as well as enough amount of data samples from social media. Here is a snippet of a sample data from the LIAR dataset in Fig. 1.

id	1123.json
label	false
statement	Health care reform legislation is likely to ma...
subjects	health-care
speaker	blog-posting
speakerjob	NaN
stateinfo	NaN
partyaffiliation	none
barelytrue	7
false	19
halftrue	3
mostlytrue	5
pantsonfire	44
context	a news release

Fig.1. Asnippet of the dataset before cleaning(preprocessed)

The credit history of each sample is also included as the previous imprecise statements for individual speaker. One example is like for speaker, Mitt Romney has this vector {19,32,34,58,33} which represents his credit history, that tells his counts for "pants on fire", "false", "barely true", "half true", "mostly true" historical statements[4]. This information tells the number of times the speaker has told true, false, pants-on-fire or any label from available ones. This dataset includes different types of features, the statement and context features are in the textual forms, few of them are in the categorical forms and remaining are numerical valued. Relevant transformation is done to each category of features for converting them to a suitable form that is accepted by the machine learning algorithms.

4. **Methodology**

We have done pre-processing on the dataset such as all the NULL valued rows are dropped. The textually based columns from the dataset are first n-gram converted and then into TFIDF vectors. These vectors are then used in finding out the Cosine similarity between each sample for the relevant column.

4.1. Textual Features extraction

Textual features extraction techniques that at used in this work are

1. Bag of words: Words from sentences are tokenized and put into bags/groups marking the token and its count.
2. N-grams: Unigrams and bigrams are extracted from the bag of words to overcome the word length formation.
3. TFIDF: Statements of tokenized words are converted into sparse matrices using TFIDF. TFIDF is calculated by calculating the Term Frequency and Inverse Document Frequency.

$$\textit{Term Frequency, } tf(t,s) = \frac{f_s(t)}{max_{\,w \in s}\, f_s(w)} \qquad (1)$$

$$\textit{Inverse Document Frequency, } idf(t, D) = ln\left(\frac{|D|}{|\{d \in D : t \in s\}|}\right) \qquad (2)$$

$$\text{Term Frequency Inverse Document Frequency, } \mathrm{tfidf(t, s, D)} = \mathrm{tf(t, s)} * \mathrm{idf(t, D)} \qquad (3)$$

Where t denotes the terms, s denotes each document in equations 1,2,3, and D denotes the corpus of documents in equation 2,3. □s(□) denotes the frequency of term t in document d in equation 1.

Cosine Similarity: Cosine Similarity is calculated from the vectors that we got from the TFIDF. This metric gives us the contextual similarity between two documents.

$$\textit{Similarity (A, B)} = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad (4)$$

Where A and B represents to TFIDF sparse matrices in equation 4.

4.2. Categorical Features

The categorical columns are label encoded and some are One hot encoded where every necessary to have better accuracy.

- Label Encoding: Categorical features are given numerical notations
- Hot Encoding: Multi categorical features are converted into binary classification by given each category a binary value.

4.3. Numerical Features

The Numerical columns are scaled and then normalized using Min-Max scaling.

- Scaling: Scaling is a process which is used to standardize the range numeric columns in dataset. [14].
- Normalization:Normalization is a process in which numeric data is normalized to the range between 0 and 1. This is done to have appropriate weight for each features of data. [14].

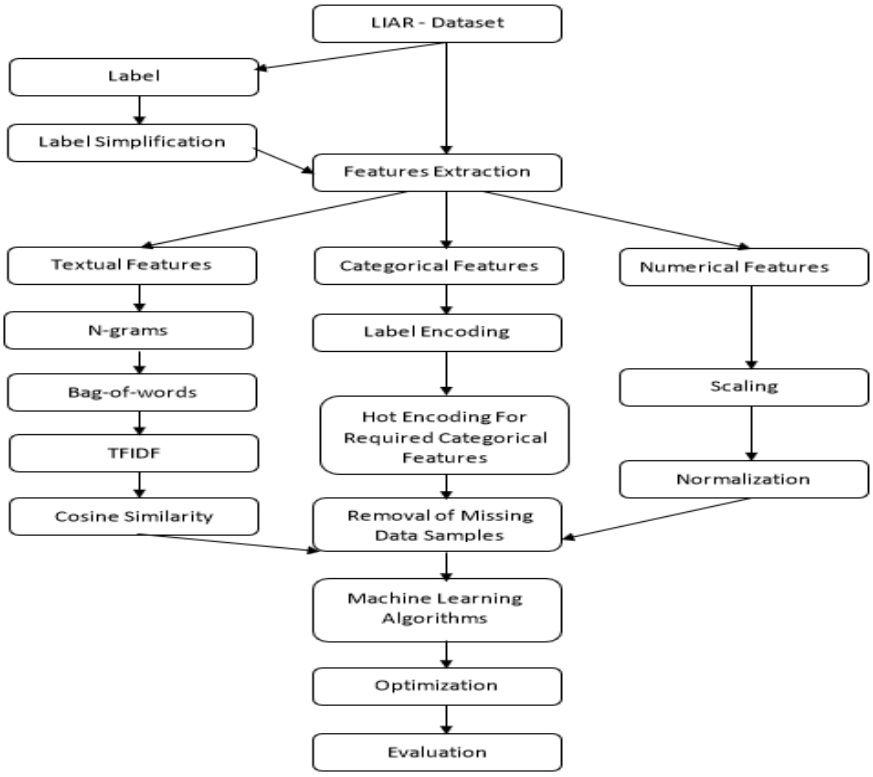

Fig. 2.Process flow chart

4.4. Ensemble Technique

Ensemble machine learning refers to a technique that integrates output from multiple learners and is applied to a dataset to make a prediction. These multiple learners are usually referred to as base learners. When multiple base models are used to extract predictions that are combined into one single prediction, that prediction is likely to provide better accuracy[15] than individual base learners. Ensemble models are known for providing an advantage over single models in terms of performance. We can ensemble models with algorithms from the same family or opt to pick them from different families.
The overall flow of the process is given in Fig 2. In this work, we start by using LIAR dataset and then doing the preprocessing steps for textual, categorical, and numerical features respectively. After preprocessing steps, the dataset is split into 80:20 ratio for training and testing phase of model in machine learning. The models are fitted on training portion and the parameters are optimized for the model by validating the model on test portion of the dataset. The same process is also followed after converting the dataset from 6 labels to 2 labels.

5. **Evaluation**

5.1. Experimental setup

We have run our evaluation on a system having a 4 core Intel CPU with 18GB of RAM. We present our result in table 1 which show metrics for precision,f1-score and recall of various machine learning algorithm on our feature extracted dataset with cross-validation done while optimized the parameters of various algorithms.We use precision, f1-score and recallas our evaluation metrics.

Precision: It is the number of correct positive by the number of positives.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Recall: It is the number of correct positive by the number of relevant samples (all samples that should have been classified positive)

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

F1-Score: It combines Precision and recall and is given as

$$F1 = 2 * \frac{precision * recall}{precision + recall} \quad (7)$$

Where TP represents True Positives, FP represents False Positives and FN represents False Negatives.

6. **Results**

Bagging Classifier and XGBoost achieve 39% accuracy in precision, recall, and F1-score in Table 1. After reclassification of the problem statement and rerunning the models on modified and feature extracted dataset, the results are presented in Table 2

Table 1.Evaluation before binary classification (in %)

Models	Precision	Recall	F1-Score
XGBoost	39	39	39
Bagging	39	39	39
RandomForest	37	37	37
ExtraTrees	37	37	37
GradientBoost	36	36	36

Table 2. Evaluation after binary classification (in %)

Models	Precision	Recall	F1-Score
Bagging	70	70	70
AdaBoost	70	70	70
RandomForest	65	65	65
ExtraTrees	62	62	62
XGBoost	62	62	62

Here Bagging Classifier and AdaBoost attain 70% accuracy in precision, F1-Score and recall. These are confirmed by running the model for 150 iterations.

7. Conclusion

In this paper, we present the task of automatic detection of fake news. We have used a new publicly available fake news dataset the LIAR-dataset. The classification of fake news from the real news is very crucial task nowadays. It is becoming an imminent threat in some situation to be not able to discern real and fake news. Our best performing models achieved accuracies that are comparable to the human ability to spot fake content.

REFERENCES

1. https://www.engadget.com/2018/12/10/more-people-get-news-from-social-media-than-newspapers/
2. https://qz.com/india/1459818/bbc-study-shows-how-indians-spread-fake-news-on-whatsapp/
3. https://scroll.in/latest/838693/fake-news-is-a-concern-for-83-of-indian-media-consumers-reveals-bbc-study
4. William Yang Wang. 2017. "Liar, liar pants on fire": A new benchmark dataset for fake news detection. In Proc. of ACL.
5. V. L. Rubin, N. J. Conroy, and Y. Chen, "Automatic deception detection: Methods for finding fake news," Proceedings of the Association for Information Science and Technology, vol. 52, no. 1, pp. 1–4, 2015.
6. Rubin, V., Conroy, N. & Chen, Y. (2015)A. Towards News Verification: Deception Detection Methods for News Discourse. Hawaii International Conference on System Sciences.
7. R. Mihalcea and C. Strapparava, "The lie detector: Explorations in the automatic recognition of deceptive language," in Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 309–312, Association for Computational Linguistics, 2009.
8. S.Feng,R.Banerjee,andY.Choi,"Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-vol 2, pp. 171–175, Association for Computational Linguistics, 2012.
9. Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic Detection of Fake News. arXiv preprint arXiv:1708.07104 (2017).
10. Olivieri, A., Shabani, S., Sokhn, M., &Cudré-Mauroux, P. (2019, January). Creating Task-Generic Features for Fake News Detection. In Proceedings of the 52nd Hawaii International Conference on System Sciences.
11. B. Pang, L. Lee, et al., "Opinion mining and sentiment analysis," Foundations and Trends® in Information Retrieval, vol. 2, no. 1–2, pp. 1–135, 2008.
12. B. D. Horne and S. Adali, "This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," arXiv preprint arXiv:1703.09398, 2017.
13. S. Gilda, "Evaluating machine learning algorithms for fake news detection," in Research and Development (SCOReD), 2017 IEEE 15th Student Conference on, pp. 110–115, IEEE, 2017.
14. https://en.wikipedia.org/wiki/Feature_scaling
15. Dietterich, T. G. (2002). Ensemble learning. The handbook of brain theory and neural networks, 2, 110-125.