Fake news detection using NLP

Karuna Gujar

April 19, 2020

Abstract

With the growth of digital world, news from all over the globe is easily and instantly available over the internet thus increasing the usage of e-news rapidly. However, many people are too credulous and fail to investigate the veracity of their news sources. This has led to the spread of fake news that misleads readers and manipulates their interests for political or commercial gain. In this work I will show the application of natural language processing tools in detecting fake news articles. An online dataset comprising of fake and real news article is used for training a classifier to discriminate between the two types. Different approaches like bag-of-words, word2vec and tf-idf were used for classification task achieving accuracy ranging between 80 % and 95 %.

1 Introduction

Fake news, also known as junk news, pseudo-news, or hoax news, is a form of news consisting of deliberate disinformation or hoaxes spread via traditional news media (print and broadcast) or online social media. Digital news has brought back and increased the usage of fake news. Fake news is written and published usually with the intent to mislead in order to damage an agency, entity, or person for financially or politically gains. Such articles often use sensationalist, dishonest, or outright fabricated headlines to increase readership.

With the enumerable resources available on the internet for the users to read news, authenticating the source of the news is challenging. Also, true and legitimate articles get lost in the deluge of fake news. According to buzzfeed, during the last three months of the 2016 U.S. presidential campaign, of the top twenty fake election-related articles on Facebook, seventeen were anti-Clinton or pro-Trump. Facebook users interacted with them more often than with stories from genuine news outlets. Thus, the problem of fake news has become a huge challenge that needs to be addressed in order to stop the spread of disinformation.

In this work, a set of natural language processing techniques were used to classify a dataset of news articles into fake and real. Different experiments were performed using a combination of different features and classifiers. The different approaches were compared using standard classification metrics.

2 Related Work

The problem of "fake news detection" seems to be fairly new and a challenging problem in natural language processing (NLP). Kareem and Awan applied different machine learning (ML) classifiers to text analysis-based vectorization methods like bag-of-words (BoW) and term-frequency-inverse-document-frequency (TF-IDF) [6]. They compared different supervised machine learning classification algorithms like logistic regression (LR), support vector machine (SVM), k-nearest neighbor (KNN), random forest classifier (RFC), naïve bayes (NB) and decision tree classifier (DTC). They collected the data from available online news websites and resources and manually annotated it as fake and real. They observed 69 % accuracy with LR and TF-IDF whereas KNN performed well with BoW and gave 70 % accuracy.

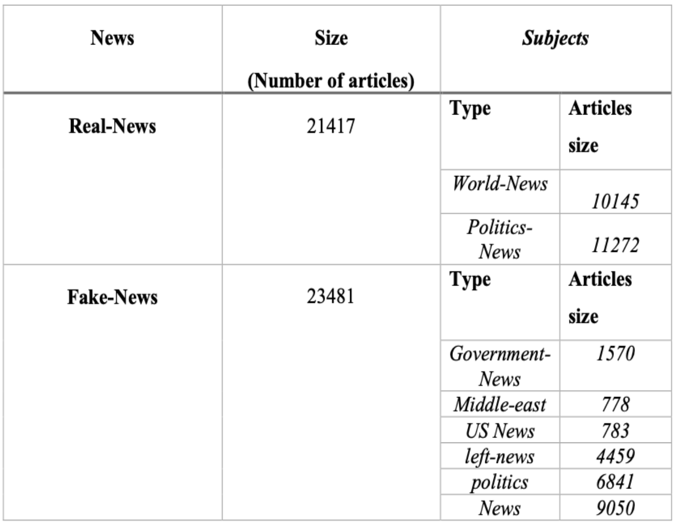
Asaad et. al. used BoW, TF-IDF and bi-gram frequency as features for ML algorithms including multinomial naive bayes (MNB) and support vector machine classifiers(LSVC). They reported that LSVC classifier performed well with the TF-IDF model as compared to MNB and that representing documents through weighted terms vectors (TF-IDF vectors) is more efficient in the case of linear classification. They also concluded that both the classifiers gave the same accuracy score with the BoW model. On the other hand, the MNB classifier gave a better score accuracy in case of bi-gram model.

Bourgonje et. al. in their study developed method for the detection of fake news based on clickbait titles [4]. They determined the relation between the news headline and its content(body) and classified whether they were related or unrelated. To achieve this, they used a dataset provided by http://www.fakenewschallenge.org with titles and content. They used CoreNLP Lemmatizer to verify the similarity between the headline of a news and it's body. The similarity score is calculated based on the n-gram technique, by studying the matches between the headline and the content. They also used TF-IDF to calculate the matching score.

3 Database

The dataset used in this study is available at `https://www.uvic.ca/engineering/ece/isot/assets/docs/ISOT_Fake_News_Dataset_ReadMe.pdf`. It contains tagged fake and real news articles. According to the authors, the truthful articles were obtained by crawling articles from Reuters.com (news website) and fake news articles were collected from unreliable websites that were flagged by Politifact (a fact-checking organization in the USA) and Wikipedia [2, 1]. The dataset contains articles on different topics, however, the majority of articles focus on political

and world news.
There are two csv files comprising the dataset - “True.csv” and “Fake.csv” each set containing about 12,600 articles. Each article contains the following information: article title, text, type and the date the article was published on. The authors cleaned and processed the data, however, the punctuation and mistakes that existed in the fake news were kept in the text. Figure 1 gives a breakdown of the categories and number of articles per category.



News	**Size** **(Number of articles)**	***Subjects***	
Real-News	21417	**Type**	**Articles size**
		World-News	*10145*
		Politics-News	*11272*
Fake-News	23481	**Type**	**Articles size**
		Government-News	*1570*
		Middle-east	*778*
		US News	*783*
		left-news	*4459*
		politics	*6841*
		News	*9050*

Figure 1: Breakdown of the categories and number of articles per category [3].

4 Research Method

The data was pre-processed by conversion to lower case, removal of stopwords and lemmatization. The data was cleaned further to remove any references to source, twitter handles (abundant in fake news) and peculiar patterns unique to either fake or real articles. Examples of such peculiarities include fake articles ending with text like ”Feature images credits to author Getty Images” or real articles starting with location and service eg. ”Washington (Reuters)”. Details of cleaning step are as follows:

1. Removal of Twitter handles: Lot of twitter handles appeared in the fake news and in order to keep the data uniform for both the categories the twitter handles were removed.
2. Removal of URLs: URLs affect the tokenization accuracy and hence were removed during processing.
3. Removal of news source and location: Fake news had news or image source at the end of every news which was ”Getty images”. Most of the true news in the dataset had ’Reuters’ appearing as the source of the news and news location was present at the beginning of the each news. In order to avoid any of these to affect the features, regular expressions were used to remove them.
4. Removal of string ’Donald Trump’: Most of these news articles dated from year 2016 and 2017. Lot of fake news had subject ’Donald Trump’ in it. In order to not let this fact affect the experiment accuracy, ’Donald’ and ’Trump’ tokens were removed.

Features were extracted from the pre-processed dataset and models were trained for classifying dataset into fake and real news. The dataset was split into train and test using a ratio of 0.8:0.2. Features were generated using enrichment and no enrichment as described below:

- **Enrichment**

  Machine learning algorithms automatically extract knowledge but studies have shown that their success is dependent on the quality dataset [5].The performance can be negatively impacted if the dataset contain extraneous or irrelevant information. Feature subset selection or enrichment can produce better generalizable models and require less computational resources.

  In this approach, 2000 features were used after enrichment from the train dataset. Enrichment was performed by using uni-grams which are more likely to be observed in real compared to fake articles and vice-versa. Word enrichment was performed by computing odds-ratio using the following formula :

$$\frac{p_1/(1-p_1)}{p_2/(1-p_2)} = \frac{p_1/q_1}{p_2/q_2} = \frac{p_1 q_2}{p_2 q_1},$$

where P_1 and P_2 is the probability of the event in each of the groups.

Top 1000 uni-grams with highest odds ratio of observation in fake vs real articles were used. Similarly, top 1000 uni-grams with highest odds ratio of observation in real vs fake articles were selected.

Features were then generated for the enriched words using the below mentioned natural language processing methods.

- **No Enrichment**

  In the second approach, no enrichment was performed and all words were used for generating features using the above mentioned natural language processing methods. Since the size of the current dataset is not a limiting factor for computation, the whole corpus was also used to train classifiers for real-fake articles. However, the features need to be generated only from the training dataset followed by transformation of test dataset.

Features were developed for enriched and non-enriched uni-grams using the following natural methods:

1. Bag of words - A binary representation of uni-grams was generated and used to train the classifiers.
2. TF-IDF - Term frequency and inverse document frequency for uni-grams was used to train a classifier using logistic regression to discriminate between real and fake articles.
3. Word2Vec - The vector representation of uni-grams from GloVe were used to train a logistic regression model. The main limitation of this method is that we can only use vector representation for known words in the corpus. The matrix for training/testing the classifier is obtained using the dot product of binary bag-of-words representation and matrix of GloVe vectors.

The features described above were used to train classifiers to distinguish real articles from fake articles. The following algorithms were used for training machine learning models:

1. Naive Bayes
2. Logistic Regression

- **Libraries**
  The following libraries were used in the current study:
  1. Scikit-learn (`https://scikit-learn.org/stable/`): Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. The library is focused on modeling data. All the classifiers discussed above will be used from this library.
  2. NLTK (`https://www.nltk.org`): This toolkit is one of the most powerful NLP libraries. Most of the feature extraction will be done using NLTK.
  3. Gensim (`https://radimrehurek.com/gensim/index.html`): Gensim is an open source Python library for natural language processing, with a focus on topic modeling. I used GloVe module from this library.

5 Results and analysis

- Metrics and Evaluation The different approaches described in the methods section were compared on the basis of the following metrics:
  1. Accuracy: the percentage of news that were predicted with the correct tag. Thus higher the accuracy, better is the result.
  2. Precision: The percentage of examples the classifier got right out of the total number of examples that it predicted for a given tag. A good model will have high values of precision.
  3. Recall: The percentage of examples the classifier predicted for a given tag out of the total number of examples it should have predicted for that given tag. Higher the recall, better the model performance.
  4. F1 Score: the harmonic mean of precision and recall.
  5. Receiver operating characteristic curve (ROC): It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0. Put another way, it plots the false alarm rate versus the hit rate. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. When AUC is approximately 0.5, model has no discrimination capacity to distinguish between positive class and negative class. Figure 2a and 2b shows the ideal ROC curve and practically achievable ROC curve respectively.

1
ROC
TPR
0
0
FPR
1
1
ROC
TPR
0
0
FPR
1

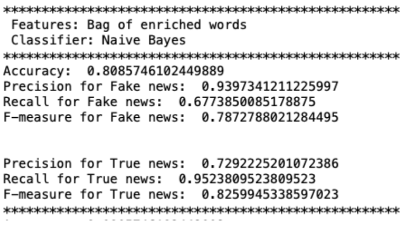
(a) Ideal ROC curve

(b) Practical ROC curve

Figure 2: A figure with two

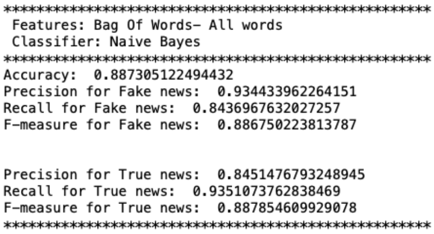
Following are the results of the experimentation performed:

1. Bag of words



```
******************************************************
 Features: Bag of enriched words
 Classifier: Naive Bayes
******************************************************
Accuracy:  0.8085746102449889
Precision for Fake news:  0.9397341211225997
Recall for Fake news:  0.6773850085178875
F-measure for Fake news:  0.7872788021284495


Precision for True news:  0.7292225201072386
Recall for True news:  0.9523809523809523
F-measure for True news:  0.8259945338597023
******************************************************
```

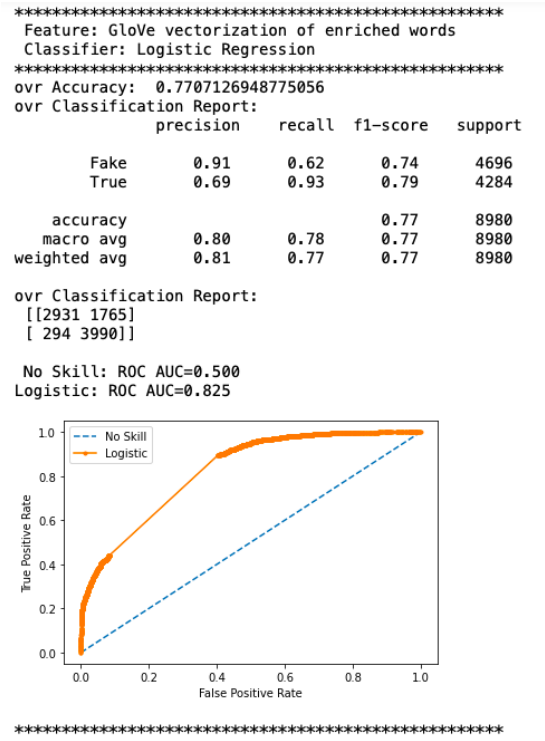
(a) Bag of enriched words



```
******************************************************
 Features: Bag Of Words- All words
 Classifier: Naive Bayes
******************************************************
Accuracy:  0.887305122494432
Precision for Fake news:  0.934433962264151
Recall for Fake news:  0.8436967632027257
F-measure for Fake news:  0.886750223813787


Precision for True news:  0.8451476793248945
Recall for True news:  0.9351073762838469
F-measure for True news:  0.887854609929078
******************************************************
```

(b) Bag of all words

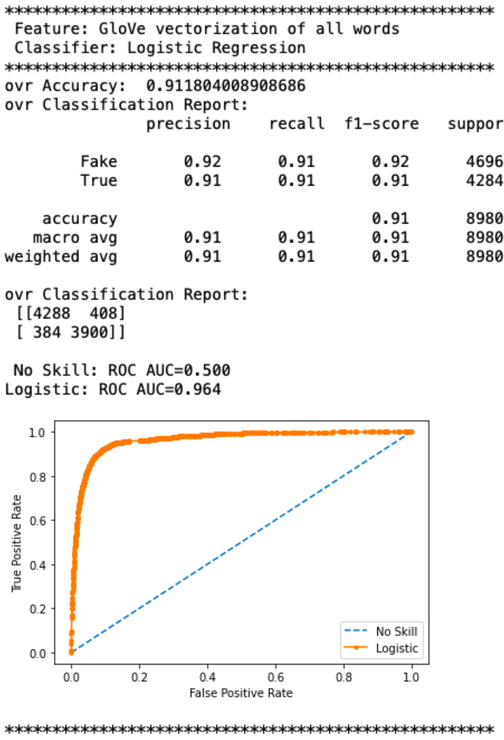
2. Word2Vec



```
******************************************************
 Feature: GloVe vectorization of enriched words
 Classifier: Logistic Regression
******************************************************
ovr Accuracy:  0.7707126948775056
ovr Classification Report:
               precision    recall  f1-score   support

        Fake       0.91      0.62      0.74      4696
        True       0.69      0.93      0.79      4284

    accuracy                           0.77      8980
   macro avg       0.80      0.78      0.77      8980
weighted avg       0.81      0.77      0.77      8980

ovr Classification Report:
 [[2931 1765]
 [ 294 3990]]

 No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.825
```

```
******************************************************
```

(a) GloVe of enriched words

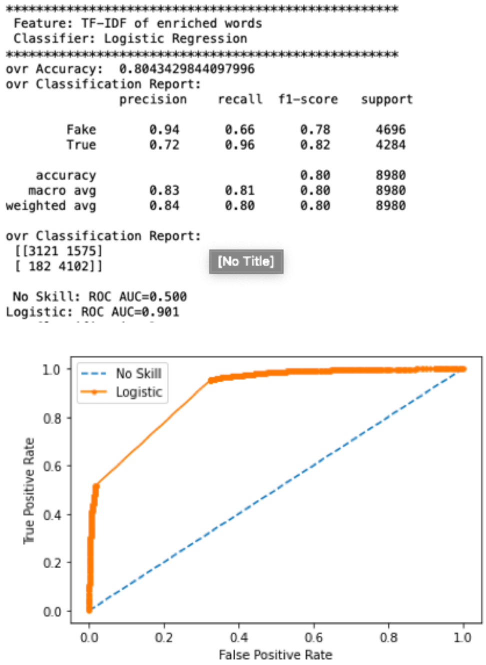


```
******************************************************
 Feature: GloVe vectorization of all words
 Classifier: Logistic Regression
******************************************************
ovr Accuracy:  0.911804008908686
ovr Classification Report:
               precision    recall  f1-score   support

        Fake       0.92      0.91      0.92      4696
        True       0.91      0.91      0.91      4284

    accuracy                           0.91      8980
   macro avg       0.91      0.91      0.91      8980
weighted avg       0.91      0.91      0.91      8980

ovr Classification Report:
 [[4288  408]
 [ 384 3900]]

 No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.964
```

```
******************************************************
```

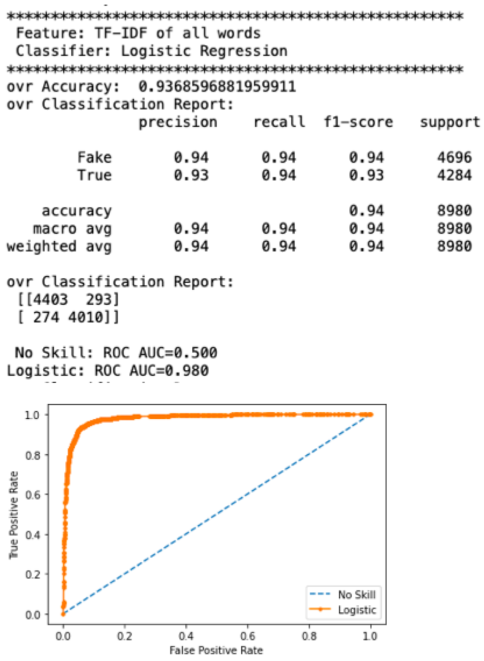
(b) GloVe of all words

3. TF-IDF



```
*****************************************************
 Feature: TF-IDF of enriched words
 Classifier: Logistic Regression
*****************************************************
ovr Accuracy:  0.8043429844097996
ovr Classification Report:
               precision    recall  f1-score   support

        Fake       0.94      0.66      0.78      4696
        True       0.72      0.96      0.82      4284

    accuracy                           0.80      8980
   macro avg       0.83      0.81      0.80      8980
weighted avg       0.84      0.80      0.80      8980

ovr Classification Report:
 [[3121 1575]
 [ 182 4102]]

 No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.901
```

(a) TF-IDF of enriched words



```
*****************************************************
 Feature: TF-IDF of all words
 Classifier: Logistic Regression
*****************************************************
ovr Accuracy:  0.9368596881959911
ovr Classification Report:
               precision    recall  f1-score   support

        Fake       0.94      0.94      0.94      4696
        True       0.93      0.94      0.93      4284

    accuracy                           0.94      8980
   macro avg       0.94      0.94      0.94      8980
weighted avg       0.94      0.94      0.94      8980

ovr Classification Report:
 [[4403  293]
 [ 274 4010]]

 No Skill: ROC AUC=0.500
Logistic: ROC AUC=0.980
```

(b) TF-IDF of all words

The results show that TF-IDF on all the words with logistic regression classifier gave the best result with 95.23% accuracy and area under the curve as high has 0.98. The TF-IDF model on all words performed best followed by Word2Vec and BoW model in that order. This is an expected result as other researchers have found TF-IDF works better than BoW. I think the Word2Vec performed worse that TF-IDF because Word2Vec is using only a subset of words that are part of the English vocabulary. Fake news articles contain a lot of unconventional words/profanities like 'nothingburger'. A possible solution would be to retrain the Word2Vec model with the whole dataset and using the retrained model for classification.

I applied a simplified version of feature selection/feature enrichment concept used in machine learning to this [7]. Here I used odds ratio instead of chi-square method. Feature selection is showed to improve results from machine learning algorithms. However, in this work using the whole corpus of words seems to work much better than a subset of words. However, there is a danger of algorithm not generalizing when all words are used. I have tried to circumvent this problem by using features that are common in the training and test dataset i.e words that are missing in training set are excluded from classifier training. Even as the features were computed independently of test dataset, we see high accuracy indicating robustness of our models.

6 Conclusion and future work

In this work several methods were evaluated for classifying fake news from real news. The accuracy achieved for classification ranged from 80% to 94%. We are easily able to achieve high accuracy levels for this dataset which suggests that there are certain identifiable elements that are good discriminators between real and fake news. TF-IDF model showed better perofrmance over others by achieving accuracy 94%.

We can further generalize these models and improved by removing names of people and entities (organizations) from the dataset by applying name entity recognition module of NLTK.

References

[1] Hadeer Ahmed, Issa Traoré, and Sherif Saad. Detection of online fake news using n-gram analysis and machine learning techniques. In *ISDDC*, 2017.

[2] Hadeer Ahmed, Issa Traoré, and Sherif Saad. Detecting opinion spams and fake news using text classification. 2018.

[3] Saad S. Ahmed H, Traore I. Isot fake news dataset, 2017.

[4] Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 84–89, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[5] Mark A. Hall and Lloyd A. Smith. Practical feature subset selection for machine learning, 1998.

[6] I. Kareem and S. M. Awan. Pakistani media fake news classification using machine learning classifiers. In *2019 International Conference on Innovative Computing (ICIC)*, pages 1–6, Nov 2019.

[7] Raghavan Prabhakar Manning D. Christopher and Schütze Hinrich. *Introduction to Information Retrieval.* Cambridge University Press, 2008.