International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 3 | Mar-Apr 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

Era of Sociology News Rumors News Detection using Machine Learning

Chandni Jain[1], S. Vignesh[2]

[1]MCA Student, [2]Assistant Professor
[1,2]Jain (Deemed-To-Be-University) Bangalore, Karnataka, India

How to cite this paper: Chandni Jain | S. Vignesh "Era of Sociology News Rumors News Detection using Machine Learning" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-3, April 2019, pp.1801-1804, URL: https://www.ijtsrd.com/papers/ijtsrd23534.pdf

IJTSRD23534

ABSTRACT

In this paper we have perform the political fact-checking and fake news detection using various technologies such as Python libraries , Anaconda , and algorithm such as Naïve Bayes, we present an analytical study on the language of news media. To find linguistic features of untrustworthy text, we compare the language of real news with that of satire, hoaxes, and propaganda. We are also presenting a case study based on PolitiFact.com using their factuality judgments on a 6-point scale to prove the feasibility of automatic political fact-checking. Experiments show that while media fact-checking remains an open research issue, stylistic indications can help determine the veracity of the text.

1. INTRODUCTION

Words in news media and political discourse have significant power to shape the beliefs and opinions of people. As a result, to maximize impact, their truthfulness is often compromised. Recently, fake news has captured worldwide interest, and since 2014, the number of organized efforts devoted solely to fact-checking has nearly tripled.1 Organizations such as PolitiFact.com are actively investigating and rateing the veracity of public figures, journalists, and organizations comment. Figure 1 shows examples of PolitiFact's quotations for truthfulness. The statement is true as stated in the first example, thoug h only because the speaker hedged its meaning with the q uantifier just. Two correlated events in the second examp le – Brexit And Google's search trends–ambiguously presented as if they were directly linked. Importantly, as in the examples above, most fact-checked statements on PolitiFact are classified as neither completely true nor completely false. Analysis indicates that falsehoods often result from subtle phrasing differences rather than straightforward manufacturing (Rubin et al., 2015)[5]. Compared with most previous literature on deception focusing on binary categorization of truth and deception, political fact-checking poses a new challenge as it involves a graded notion of truthfulness.

In this paper we present an analytical study that characterizes the language of political quotes and news media written with varying degrees of truth and intentions. We also investigate graded deception detection, using the political fact-checking database available at PolitiFact to determine the truthfulness on a 5-point scale.

2. ANALYSIS

We sampled standard trusted news articles from the English Gigaword corpus and crawled articles from seven different unreliable news sites of different types to analyze linguistic patterns across different types of articles. Table 1 shows sources identified under each type according to the U.S. News & World Report.

Nature of news	Reference	No. of doc	No. of tokens
TRUE	Gigaword news	13,900	571
IRONY	The Borowitz	656	250
FAKE	American News	6917	200
LITERATURE	Activist report	17,669	1,111

Table 1

These types of news include:

- Irony: imitates real news but still points out to the reader that it is not intended to be taken seriously

- Deception: convinces readers of the validity of a story fuelled by paranoia
- Literature: misleads readers to believe in a particular political / social agenda Unlike hoaxes and propaganda.

Falsehoods and satire tend to invent stories, whereas propaganda often combines truths, falsehoods and ambiguities to confuse readers. We applied various lexical resources to confident and fake news articles in order to characterize differences between news types. In communication theory and stylistic analysis in computational linguistics, we draw lexical resources from past works. We Tonkenized NLTK text, compute the count per document and report averages per article of each type for each lexicon (Bird et al., 2009). The LIWC, a lexicon widely used in social science studies, has been among the first to be found in these lexicons (Pennebaker et al., 2015). Furthermore, we estimate that words with lexicogenous sentiment are strongly and weakly subjective (Wilson et al., 2005). In the drama or sensationalization of a news story, subjective words may be used. In addition, we use lexicons to hedge (Hyland, 2015) as hedging indicates a vague, obscure language. Finally, we're introducing intensifying lexicons we've been crawling from Wiktionary based on a hypothesis that fake news articles are trying to animate stories to attract readers.

Five Lists of Wiktionary Words have been compiled, which involve a certain degree of dramatization (comparatives, superlatives, action adverbs, adverbs and modal adverbs).

Argument: the ratio of averages for a handful of the measured features between unreliable news and true news. Ratios greater than one denote features more prominent in fake news and features more prominent in truthful news features ratios less than one denote. After Bonferroni correction, the ratios between reported unreliable / reliable news are statistically significant ($p<0.01$) with Welsch t-test. Our results show that pronouns of the first person and second person are used more in less reliable or disappointing types of news. This contrasts studies in other fields (Newman et al., 2003), which found fewer self-references in people telling their personal views lies, In contrast to that domain, news writers try to be indifferent. Editors at trusted sources may be more Rigorous removal of seemingly too personal language, which is one reason why this result differs from other detection domains. Instead, this finding confirms previous work in written domains found by Ott et al. (2011) [5] and Rayson et al. (2001), which found such pronouns to be indicative of imaginative writing. Maybe imaginative storytelling domains are closer to detecting unreliable news than detecting views. Furthermore, our results show that words that can be used to amplify – subjectives, superlatives, and modal adverbs –are all more used by false news. In truthful news, words used to offer concrete figures –comparisons, money, and numbers – appear more.

3. RELATED WORK

There are mainly Four Modules:

Classifier.py: We've built all the classifiers here to predict the detection of fake news. The extracted characteristics are fed into various classfiers. We used sklearn Naive-bayes, logistic regression, linear SVM, decent Stochastic gradient, and random sklearn forest classifiers. Finally selected model was used with the probability of truth for fake news detection. In addition to this, we also extracted from our term-frequency tfidf vectorizer the top 50 features to see which words are most important in each of the classes. We also used Precision-Recall and learning curves to see how the training and test set works when we increase the data in our classifiers.

DataPrep.py: This file contains all the necessary pre-processing functions for processing all documents and texts input. First we read the data files for train, test and validation, then we didsome pre-processing such as tokenizing, stemming, etc. There are some analyzes of exploratory data such as distribution of variable responses and checks of data quality such as null or missing values etc.

FeatureSelection.py: We performed extraction of features and selection methods from sci-kit learning python libraries in this file. We used methods such as simple bag-of-words and n-grams for selection of features and then term frequency such as tf-tdf weighting.

Prediction.py: Our finally selected and best performnig classifier was Logistic Regression, which was then saved with the final model.sav name on the disk. Once this repository is closed, this model will be copied to the user's machine and used to classify the fake news by prediction.py file. It takes a news article as user input, then model is used for the final output of classification shown to the user along with the probability of truth.

Figure1 Future directions and open issues for fake news detection on social media.

4. METHODOLOGY

4.1 N-GRAM

N-grrm modeling is a popular approach to identifying and analyzing features used in the fields of language modeling and processing of natural language. N-gram is an adjacent sequence of n-length items. It could be word, byte, syllable, or character sequence. Word-based and character-based n-grams are the most commonly used n-gram models in text categorization. We use word-based n-gram in this work to represent the document context and generate features to classify the document. To distinguish between fake and honest news articles, we develop a simple n-gram-based classifier. The idea is to generate different sets of n-gram frequency profiles from the training data to represent false information And articles of truthful news. We used several word-based baseline n-gram features and looked at the effect of n-gram length on the accuracy of various classification algorithms.

4.2 Data Pre-processing

Before representing the data using n-gram and vector-based model, some refinements such as stop-word removal, tokenization, lower case, sentence segmentation, and punctuation removal need to be subjected to the data. This

will help us to reduce the actual data size by removing the data's irrelevant information[13]. To remove punctuation and non-letter characters for each document, we created a generic processing function; then we lowered the letter case in the document. In addition, a tokenizer based on a n-gram word was created to slice the text based on the length of n

Stop Word Removal

Stop words are meaningless words in a language that, when used as features in text classification, will create noise. These are words that are commonly used extensively in sentences to help connect thought or help in the structure of sentences. Articles, prepositions and conjunctions, as well as certain pronouns, are considered words of stop. We have removed common words like, a, about, an, are, like, at, be, by, for, from, how, in, is, on, or, that, the, these, this, too, was what, when, where, who, who, will, etc. These words have been removed from each document and the documents processed have been stored and passed on to the next step.

Stemming

The next step is to transform the tokens into a standard form after tokenizing the data. Stemming simply transforms the words into their original form and lowers the number of word types or classes in the data. For example, the words "Running," "Ran" and "Runner" will be reduced to the word "run." To make classification faster and more efficient, we use stemming. We also use Porter stemmer, which due to its accuracy is the most commonly used stemming algorithms.

4.3 Features Extraction

Learning from high-dimensional data is one of the challenges of categorizing text. There are a large number of terms, words, and phrases in documents that result in the learning process having a high computational burden. In addition, the classifiers ' accuracy and performance can be hurt by irrelevant and redundant features. Thus, to reduce the size of the text feature and avoid large space dimensions of the feature, it is best to perform feature reduction. Two different selection methods of features, namely Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF), were studied in this research. The following describes these methods.

A. Term Frequency (TF)

Term Frequency is an approach that uses the numbers of words in the documents to determine the similarity between documents. Each document is represented by a vector of equal length containing the counts of words. Next, each vector is standardized in a way that adds to one the sum of its elements. Then each word count is converted into the likelihood of such a word in the documents. For example, if a word is in a particular document, it is represented as one, and if it is not in the document, it is set to zero. Consequently, each document is represented by word groups.

B. TF-IDF

The Term Frequency-Inverted Document Frequency (TF-IDF) is a weighting metric that is frequently used to retrieve information and process natural language. It is a statistical metric used to measure the significance of a term in a dataset to a document. With the number of times a word appears in the document, a term importance increases,[11]but this is counteracted by the word frequency in the corpus. One of IDF's main features is weighing down the term frequency while scaling up the rare frequencies. For example, words like "the" and "then" often appear in the text, and if we only use TF, the frequency count will be dominated by terms like these. Use of IDF scales down the impact[10].

4.4 Classification Process

Figure 1 is a graphical representation of the process of classification. It begins with pre-processing the data set by removing unnecessary data characters and words. N-gram features are extracted and the documents involved are represented by a matrix of features. The final step in the process of classification is to train the classifier. To predict the class of the documents, we investigated various classifiers. Specificall six different machine learning algorithms have been investigated, namely Stochastic Gradient Descent.

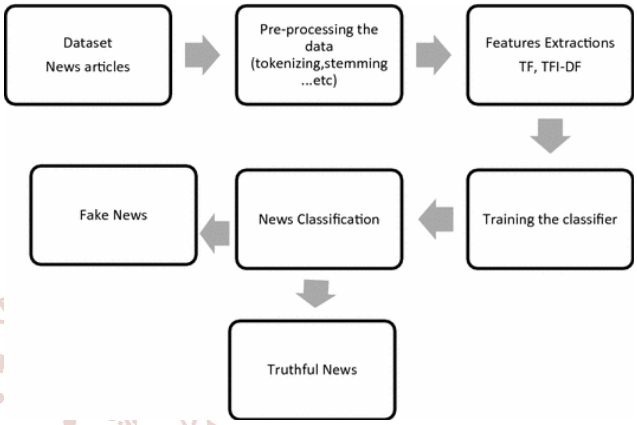

Figure 2 Classifier Workflow

We used the Python Natural Language Toolkit (NLTK) implementations of these classifiers. (SGD), Support Vector Machines (SVM), Linear Support Vector Machines (LSVM), K-Nearest Neighbor (KNN), and Decision Trees (DT).

	TF-IDF		TF		
N-gram size	10,000	50,000	1000	5000	10,000
Unigram	86.0	84,0	85.0	72.0	69.0
BiGram	78.0	73.0	53.0	47.0	53.0
TriGram	55,0	37.0	47.0	48.0	40.0
Four Gram	71.0	59,0	51.0	47.0	47.0

Table 1 SVM accuracy results. The second row corresponds to features size. Accuracy values

We divide the dataset into sets of training and testing. For example, we use 5-fold cross validation in the subsequently presented experiments, so about 80% of the dataset is used for training and 20% for testing in each validation.

5. SNAPSHOTS:

Figure 3 In this image the prediction.py file is been called to check whether the given statement is Fake or true.

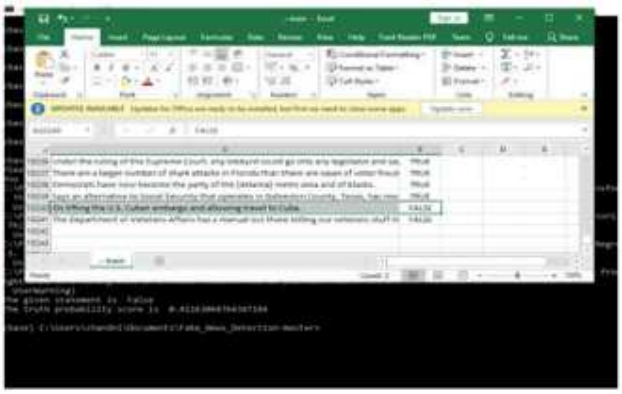

Figure 4 It shows the given statement is False as well as it gives the confusion matrix decimal value.

6. CONCLUSION

In this paper, through the lenses of various feature extraction techniques, we presented a detection model for fake news using n-gram analysis. In addition, two different extraction techniques features and six different machine learning techniques were investigated. When using unigram features and Linear SVM classifier, the proposed model achieves its highest precision. The highest precision score is 92 %. Fake news detection, with few public datasets, is an emerging research area. We run our model on an existing dataset, demonstrating that our model exceeds the original approach published by the dataset authors.The examine truthfulness and its linguistic attributes that contribute across multiple domains[4], such as online news sources and public statements. I have perform multiple predictive tasks on fact-checked statements of varying truth levels (graded deception) as well as a deeper linguistic comparison of different types of fake news such as propaganda, satire, and hoaxes. I have shown that fact-checking is indeed a challenging task, but that different lexical features can help us understand the differences between more reliable and less reliable[7] digital news sources. Fake detection of news in finely grained classes, which is also a challenging but interesting and practical issue from short statements. Hypothetically, the problem may be associated with the

problem of sarcasm detection[15]. It will therefore also be interesting to see the effect of implementing the existing methods that are effective in the field of sarcasm detection in the field of Fake News detection

7. ACKNOWLEDGEMENTS

I would like to thank LIAR LIAR PANTS ON FIRE who provide user a huge datasets which is very useful while Traning the data, feature selection of data and Test the data, apart from this I would like to thank my Project guide who guided me in all the way and made me understand the importance of machine learning concepts .

8. REFERENCES

[1] N. J. Conroy, V. L. Rubin, 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.

[2] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 171–175.

[3] Shlok Gilda, Department of Computer Engineering, Evaluating Machine Learning Algorithms for Fake News Detection,2017 IEEE 15th Student Conference on Research and Development (SCOReD)

[4] Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. 2014. News credibility evaluation on microblog with a hierarchical propagation model. In Data Mining (ICDM), 2014 IEEE International Conference on, pages 230–239. IEEE.

[5] Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

[6] Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, and Can Candan., 2016. caret: Classification and Regression Training. R package version 6.0-70.

[7] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch, 2015. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-7

[8] Wang, W. Y.: Liar, Liar Pants on fire: a new Benchmark dataset for fake news detection.

[9] arXiv preprint (2017). arXiv:1705.00648

[10] Rubin., Victoria, L., et al.: Fake news or truth? Using satirical cues to detect potentially misleading news. In: Proceedings of NAACL-HLT (2016)

[11] Gottfried, J., Shearer, E.: News use across social media platforms. Pew Res. Cent. 26 (2016)

[12] Gottfried, J., et al.: The 2016 presidential campaign– a news event that's hard to miss. Pew Res. Cent. 4 (2016)

[13] Silverman, C., Singer-Vine, J.: Most americans who see fake news believe it, new survey says. Buzz Feed News (2016)

[14] Dewey, C.: Facebook has repeatedly trended fake news since firing its human editors. Washington Post (2013)

[15] Horne, B. D., Adali, S.: This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: the 2nd International Workshop on News and Public Opinion at ICWSM (2017)

[16] Bur foot, C., Baldwin, T.: Automatic satire detection: are you having a laugh? In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, 4 August 2009, Suntec, Singapore (2009)