JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Fake News Prediction in Machine Learning

Sanket Muchhala
sanket.muchhala@gmail.com
B.E. I. T Student.
Thakur College of Engineering and Technology.

Hardik Sodhani
hardik.sodhani@gmail.com
B.E. I. T Student,
Thakur College of Engineering and Technology.

Shreeram Geedh
shreeramgeedh36@gmail.com
Associate Data Analyst,
Ugam Solutions
Mumbai, India.

Abstract:
This paper discusses the use of natural language processing techniques to identify 'false stories', that is, misleading stories from unreliable or authoritative sources. Using data obtained from Signal Media and a list of sources from OpenSources.co, using bi-grams term frequency-inverse document frequency (TF-IDF) and free language grammar (PCFG) access to a total of approximately -11,000.. We test our database on multi-phase algorithms Vector Support Machines, Stochastic Gradient Decreases, Gradient Enlargement, Determined Decision Trees, and Random Forests. We found that the TF-IDF bi-grams included in the Stochastic Gradient Descent model identifies unreliable sources with 77.2% accuracy, and PCFGs have little effect on memory.

Index Terms **– ML, Decision Tree, Prediction, supervised learning.**

I. INTRODUCTION

In 2016, the rise of disinformation amidst American political rhetoric became a major issue, especially following the election of President Trump [1]. The term 'fake news' has become a common word in the story, especially to describe misleading articles and misleading facts published primarily for the purpose of making money from page views. In this paper, we want to produce a model that can accurately predict the chances that a given article is a false story.

Facebook has been the site of much criticism following media attention. They have already used the feature for users to flag fake news on the site [2]; however, it is clear from their public declarations that they are diligently researching their ability to classify these topics automatically. Indeed, it is not an easy task. The algorithm provided should be politically neutral - as false news exists on both ends of the spectrum - and provide equal balance to official media sources across the spectrum. Moreover, the question of legitimacy is difficult. We need to decide what makes a new site 'legitimate' and how to determine this in a straightforward way.

In this paper, we compare the performance of models using three distinct feature sets to understand what factors are most predictive of fake news: TF-IDF using bi-gram frequency, syntactical structure frequency (probabilistic context free grammars, or PCFGs), and a combined feature union. In doing so, we follow the existing literature on deception detection through natural language processing (NLP), particularly the work of Feng, Banerjee, and Choi [3] with deceptive social media reviews. We find that while bi-gram TF-IDF yields predictive models that are highly effective at classifying articles from unreliable sources, the PCFG features do little to add to the models' efficacy. Instead, our findings suggest that, contrary to the work done in [3], PCFGs do not provide meaningful variation for this particular classification task. This suggests important differences between deceptive reviews and so-called 'fake news'. We then suggest additional routes for work and analysis moving forward. Section II briefly describes the previous work done in the field of text editing and false news detection. Section III describes the database used to train separator. Section IV shows how to perform the feature and pre-processing steps. Section V describes the actual modelling process and compares the results from different algorithms. Finally, Section VI presents the conclusions and outlines the possibilities for further development in the proposed course.

Abbreviations and Acronyms

ML- Machine Learning,

2.1 Related Words

There is the subject of a lot of research on the topic of machine learning methods for fraud detection, most of which focus on separating online updates and publicly available social media posts. Especially since the end of 2016 during the US Presidential election, the question of deciding 'false news' has also become a hotly debated issue in the literature.

Conroy, Rubin, and Chen [4] point out several methods that seem promising in order to better distinguish misleading articles. They realized that simple content-related programs and shallow marking of the speech part (POS) indicated inadequacy of classification work, often failing to account for important contextual information. Instead, these methods have been shown to be useful only in conjunction with sophisticated analytical methods. An in-depth Syntax analysis using Probabilistic Context Free Grammars (PCFG) has been shown to be very important when combined with n-gram methods. Feng, Banerjee, and Choi [3] are able to achieve 85% -91% accuracy in classification-related classification activities using corpora online review.
Feng and Hirst [5] used semantic analysis looking at the pairs of 'object: descriptive' to find contradictions in the text above Feng's first deep syntax model for further development. Rubin, Lukoianova and Tatiana [6] analyzed speech structure using a vector space model with similar success.
Ciampa glia et al. [7] use networks that resemble language patterns that require an existing knowledge base.

TABLE I: Comparison of high unreliable and reliable sources on the quantity of articles.

Top Five Unreliable News Sources		Top Five Reliable News Sources	
Before It's News	2066	Reuters	3898
Zero Hedge	149	BBC	830
Raw Story	90	USA Today	824
Washington Examiner	79	Washington Post	820
Infowars	67	CNN	595

III. DATA PREPARATION

2.2 Dataset Description

Conroy, Rubin and Chen [4] outline several requirements for a helpful corpus for use in these contexts (shortened for relevance):

1. Availability of both truthful and deceptive instances.
2. Verifiability of 'ground truth'.
3. Homogeneity in lengths.
4. Homogeneity in writing matter.
5. Predefined timeframe.
6. The manner of delivery (e.g., sensational, newsworthy).

To address some of these challenges, we are posting additional chorus descriptions on the OpenSources.co [8] website which includes an ongoing list of untrue and unreliable media sources. It is far from perfect and has opponents, as any such list would likely be. Ultimately, our modelling should be an independent data source and be able to use a better chorus or organization if available.

Finding a newsletter is very difficult due to copyright issues. We have obtained a set of data published by Signal Media in conjunction with the Recent News Recovery Conference 2016 to further research the news articles [9]. The database contains approximately 1 million articles from various media outlets since September 2015. Sources include major news outlets such as Reuters [10] as well as local media outlets and blogs. From this database, we filter to include articles from reliable verified sources (labelled 0) and unreliable verified sources (labelled 1).

Our clean data contains 11051 articles. 3217 (29%) were labelled as false. Reliable articles come from 14 unique sources. Unscrupulous articles come from 61 unique sources. In false stories our examples are taken from the same source: Before the News.

2.3 Resampling to Account for Skewed Distributions

In order to limit the extent to which our models will study the differences between 'Before News' and Reuters [10], we force the distribution to cover a limited range by negligently taking samples of the largest source n small n. .
We selected 500 n-max articles for use as it seemed prudent, though not legally supported. Nor do we downgrade low frequency sources to maintain a certain range of resources. Correct n-max (or potential n-min) is a research question that is of interest to itself. Additional research methods may consider changing this number to get a better result if you are facing the same corpus difficulty. We note that before and after TABLE II: Irregular and random model performance everywhere metrics.

Model	Accuracy	Precision	Recall
Naive	67.89%	54.22%	54.22%
Random	56.42%	32.18%	32.18%

on re-distribution of the distribution, we see a slight decrease in accuracy, indicating that we have entered certain sources rather than the categories themselves with a more skewed distribution of resources.

III FEATURE GENERATION

Our approach evaluates the performance of models trained on three feature sets:
1. Bigram Term Frequency-Inverse Document Frequency.
2. Normalized frequency of parsed syntactical dependencies.
3. A union of (1) and (2).

Featured design, we rely on the Spacy Python [11] package to make tokens, mark part of speech, system classification, and named business recognition. Spacy [11] is used in Cython (a Python language superset that allows C-code to be executed in Python using the Python / C API) [12], allowing for faster performance compared to other NLP as NLTK [13]].

Several studies from peer-reviewed journals found that Spacy [11] achieves performance in the classification and recognition of business functions compared to other widely used tools, while having significant advantages in terms of speed [14]. That is why we have chosen to use Spacy [11] in addition to the many established options such as Stanford's Probabilistic Context Free Grammar Java Java. [15]

From the text of the green article, we use Spacy [11] and SciKit Learn [16] [17] to produce relevant features. We use Spacy's support [11] in multiple combinations to align the production process and feature of the SciKit Learn's Pipeline [17] to create modification and modification methods that can be used in training data and used in the test set.

3.1 Pre-processing

We rub the articles of whatever the source name means. Because the reliability / distrust classification is determined at the source level, this step is necessary to ensure that the model does not just read the layout from known sources to labels. We also remove Twitter handles and email addresses (which often appear in journalists' biographies) for the same reason.

3.2 Term Frequency-Inverse Document Frequency

The first feature set is the vectorized bigram Term Frequency Inverse Document Frequency. This is a measure of how often the phrase bigram appears in the document in relation to how often the phrase bigram appears in all documents on the corpus.
Due to the political climate of our corporation, we want to reduce model knowledge by individuals and institutions.

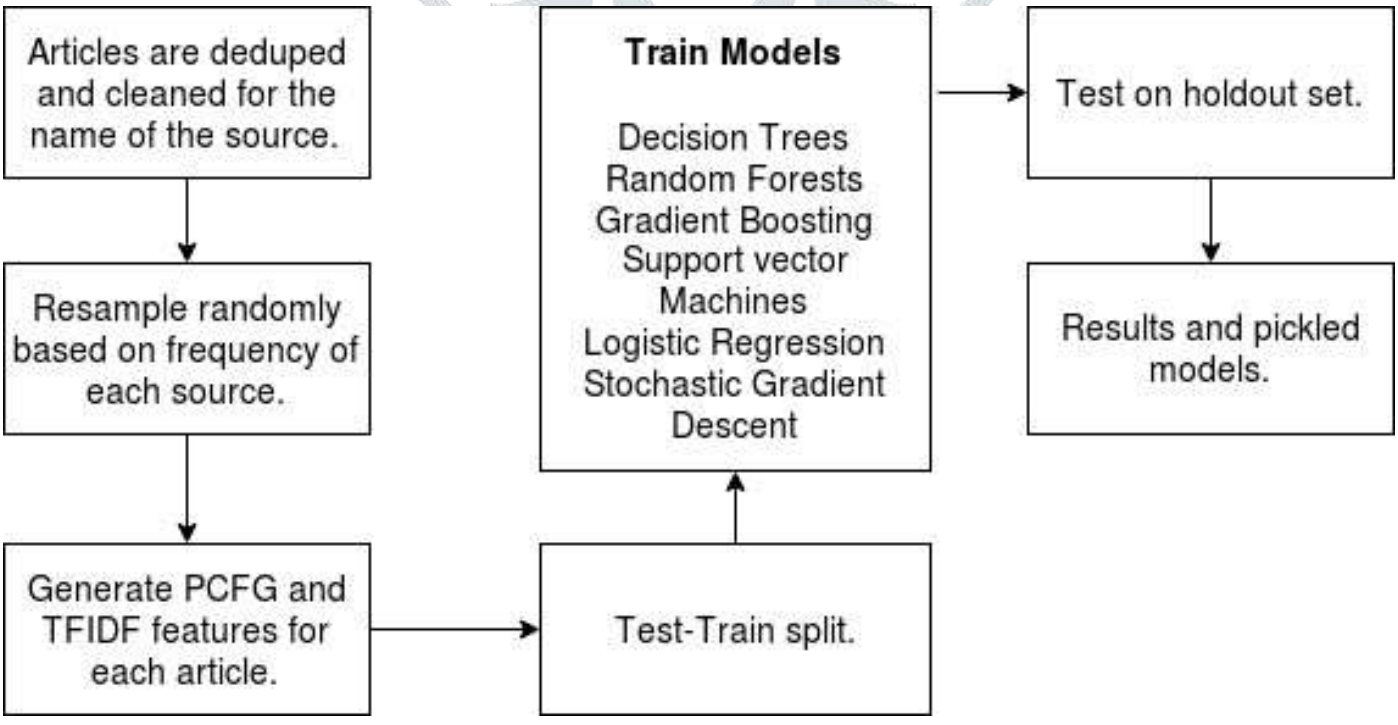

Fig. 1: Pipeline representation.

TABLE III: Average model performance with both PCFG and TF-IDF bi-gram features at 0.7 score threshold for categorization.

Model	Area Under Curve	Precision	Recall	Accuracy
Bounded Decision Trees	65.9%	66.9%	37.9%	67.6%
Gradient Boosting	75.6%	40.2%	16.1%	65.7%
Random Forests	80.0%	84.2%	18.4%	64.8%
Stochastic Gradient Descent	87.5%	74.1%	71.7%	65.7%
Support Vector Machine	84.3%	80.9%	44.5%	73.6%
Baseline	-	32.18%	32.18%	67.89%

TABLE IV: Average model performance with only TF-IDF bi-gram features at 0.7 score threshold for categorization.

Model	Area Under Curve	Precision	Recall	Accuracy
Bounded Decision Trees	50.0%	40.9%	10.8%	60.1%
Gradient Boosting	50.0%	40.9%	10.8%	60.1%
Random Forests	50.0%	40.9%	10.8%	60.1%

Model	Area Under Curve	Precision	Recall	Accuracy
Bounded Decision Trees	60.7%	58.5%	23.3%	66.1%
Gradient Boosting	79.4%	41.0%	22.3%	68.7%
Random Forests	78.8%	82.9%	25.3%	67.6%
Stochastic Gradient Descent	88.3%	88.8%	45.3%	77.2%
Support Vector Machine	85.6%	81.3%	48.1%	76.2%
Baseline	-	32.18%	32.18%	67.89%
Stochastic Gradient Descent	50.0%	40.9%	10.8%	60.1%
Support Vector Machine	50.0%	40.9%	10.8%	60.1%
Baseline	-	32.18%	32.18%	67.89%

TABLE V: Average model performance with only PCFG features, classifying the top 5% of scores as positive (k = 0.05).

mentioned in the title text. If not, we endanger the model by simply reading patterns like 'Clinton corrupt' that define the title and point of view of the text, rather than the result of interest (whether this source is trustworthy or not). Additionally, these patterns will be more sensitive to a particular media cycle. To address this concern, we are introducing a step-by-step process for token-making to use Spacy's [11] business-focused business attention to substantiate all the business pronouns in the pronoun, e.g., <-NAME> or <-ORG->.

We use SKLearn [16] to calculate the TF-IDF for each bigram within each document and to construct a separate matrix for the resulting elements. To keep our data size at a manageable size, we limit vocabulary to consider only the top 3000 words ordered by the frequency term across the chorus. We haven't explored different ways or limitations for word-for-word additions, or with different n-gram lengths, but this may be an area to be explored in future work.

3.3 Normalized Syntactical Dependency Frequency

In total, we find 3000 elements in the TF-IDF family and 46 in the program family. Names of key features are not yet available, which limits our ability to check which features of the document appear to predict its validity. This will allow us to determine better whether to classify by reading the patterns of topics or the outcome of the interest and to allow for a more complete evaluation of the strength of our results.

IV MODELING AND EVALUATING

4.1 Our Pipeline

After cleaning the data and generating features, we execute a 90/10 random test-train split on the dataset and feed it into a modelling pipeline. This pipeline iteratively fits models varying the tuning parameters with which they are executed up to 50 times, depending on the number of possible permutations for that model. These models are then tested on the 10% holdout data to understand their performance

4.2 Baseline Models for Comparison: Naive and Random

As a basic comparison to understanding the performance of our models, we look at two approaches. First, the Naive Bayes model predicts the entire mass category; in thisZ case, all the articles come from reliable sources. Second, the model randomly selects the classification of each article as reliable or unreliable based on the background opportunities of that class in the training set. These are the Naive and Random models, respectively. We describe their work in more detail in Table II.

4.3 Combining PCFG and TF-IDF bi-gram features

Combining both sets of features, our models perform well beyond our foundation as shown in Table III.

We recognize that our best models tend to be Stochastic Gradient Descent (SGD) models, which, given that they tend to perform better with smaller and larger data, are not surprising. In particular, SGDs work best in accuracy while remembering a lot, which means that these models will work well both as identifying articles that are more important than 'fake news filters.'

4.4 TF-IDF Bigram Only Model Performance

Removing PCFG features allows us to understand more deeply the value of those features in achieving these combined feature results. The results of this limited feature are shown in Table IV.

PCFG removal improves most metrics in all our models. This is surprising, indicating that PCFG features add a small amount of prediction to models. Indeed, the only significant decrease in performance is in our commemorative calculations of Tree Decisions and SGDs.

V. Conclusion and Future Scope

The results obtained above are very promising. This approach demonstrates that the frequency of the term has the potential to predict false stories - an important first step in using machine separation to identify. The most effective models with ROC AUC all Stochastic Gradient Descent models are trained with a set of TF-IDF feature only. We see that PCFGs do not add a large amount of prediction, but balance Remember our most efficient model. This indicates that PCFGs are more suitable for use of the Fake-News Filter type compared to, for example, directing fake news sites for review. The TF-IDF demonstrates promising power to predict, even if we ignore named businesses, but we remain skeptical that this approach will have the potential to change news cycles. However, this will require a complete corpus.

Aside from the high performance of our separator, there are improvements. We tested our models using absolute thresholds, which may not be the most reliable for models where the chances of getting points are not well measured. Although the TF-IDF works best, we may be overly sensitive to key topics / goals in the ongoing media cycle. Also, a vectorized approach similar to ours makes it technically difficult to identify which features are most important, thus disrupting our analysis. These problems limit our analysis and thus prevent broader fulfilment. We plan to address these issues in future work.

REFERENCES

[1] S. Maheshwari, *How fake news goes viral: A case study*, Nov. 2016. [Online]. Available: https://www.nytimes. com / 2016 / 11 / 20 / business / media / how - fake - news spreads.html (visited on 11/08/2017).

[2] A. Mosseri, *News feed fyi: Addressing hoaxes and fake news*, Dec. 2016. [Online]. Available: https://newsroom. fb. com / news / 2016 / 12 / news - feed - fyi - addressing hoaxes-and-fake-news/ (visited on 11/08/2017).

[3] 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.

[4] 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.

[5] V. W. Feng and G. Hirst, "Detecting deceptive opinions with profile compatibility.," in *IJCNLP*, 2013, pp. 338– 346.

[6] V. L. Rubin and T. Lukoianova, "Truth and deception at the rhetorical structure level," *Journal of the Association for Information Science and Technology*, vol. 66, no. 5, pp. 905–917, 2015.

[7] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PloS one*, vol. 10, no. 6, e0128193, 2015.

[8] *Opensources*. [Online]. Available: http: / / www . opensources.co/ (visited on 11/08/2017).