Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques

Hadeer Ahmed[1(✉)], Issa Traore[1], and Sherif Saad[2]

[1] ECE Department, University of Victoria, Victoria, BC, Canada
meresger.hs@gmail.com, itraore@ece.uvic.ca
[2] School of Computer Science, University of Windsor, Windsor, ON, Canada
Sherif.SaadAhmed@uwindsor.ca

Abstract. Fake news is a phenomenon which is having a significant impact on our social life, in particular in the political world. Fake news detection is an emerging research area which is gaining interest but involved some challenges due to the limited amount of resources (i.e., datasets, published literature) available. We propose in this paper, a fake news detection model that use n-gram analysis and machine learning techniques. We investigate and compare two different features extraction techniques and six different machine classification techniques. Experimental evaluation yields the best performance using Term Frequency-Inverted Document Frequency (TF-IDF) as feature extraction technique, and Linear Support Vector Machine (LSVM) as a classifier, with an accuracy of 92%.

Keywords: Online fake news · Text classification · Online social network security · Fake news detection · N-gram analysis

1 Introduction

In the recent years, online content has been playing a significant role in swaying users decisions and opinions. Opinions such as online reviews are the main source of information for e-commerce customers to help with gaining insight into the products they are planning to buy.

Recently it has become apparent that opinion spam does not only exist in product reviews and customers' feedback. In fact, fake news and misleading articles is another form of opinion spam, which has gained traction. Some of the biggest sources of spreading fake news or rumors are social media websites such as Google Plus, Facebook, Twitters, and other social media outlet [1].

Even though the problem of fake news is not a new issue, detecting fake news is believed to be a complex task given that humans tend to believe misleading information and the lack of control of the spread of fake content [2]. Fake news has been getting more attention in the last couple of years, especially since the US election in 2016. It is tough for humans to detect fake news. It can be argued that the only way for a person to manually identify fake news is to have a vast knowledge of the covered topic. Even with the knowledge, it is considerably hard to successfully identify if the information in the article is real or fake. The open nature of the web and social media in

I. Traore et al. (Eds.): ISDDC 2017, LNCS 10618, pp. 127–138, 2017.
https://doi.org/10.1007/978-3-319-69155-8_9

addition to the recent advance in computer science simplify the process of creating and spreading fake news. While it is easier to understand and trace the intention and the impact of fake reviews, the intention, and the impact of creating propaganda by spreading fake news cannot be measured or understood easily. For instance, it is clear that fake review affects the product owner, customer and online stores; on the other hand, it is not easy to identify the entities affected by the fake news. This is because identifying these entities require measuring the news propagation, which has shown to be complex and resource intensive [3]. Trend Micro, a cyber security company, analyzed hundreds of fake news services provider around the globe. They reported that it is effortless to purchase one of those services. In fact, according to the report, it is much cheaper for politicians and political parties to use those services to manipulate election outcomes and people opinions about certain topics [4, 5]. Detecting fake news is believed to be a complex task and much harder than detecting fake product reviews given that they spread easily using social media and word of mouth.

We present in this paper an n-gram features based approach to detect fake news, which consists of using text analysis based on n-gram features and machine learning classification techniques. We study and compare six different supervised classification techniques, namely, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Linear Support Vector Machine (LSVM), Decision tree (DT) and Stochastic Gradient Descent (SGD). Experimental evaluation is conducted using a dataset compiled from real and fake news websites, yielding very encouraging results.

The remaining sections are structured as follows. Section 2 is a review of related works. Section 3 introduces our proposed approach and model. Section 4 presents the experiments conducted to evaluate our proposed fake news detection model. Section 5 makes concluding remarks and discusses future work.

2 Related Works

Research on fake news detection is still at an early stage, as this is a relatively recent phenomenon, at least regarding the interest raised by society. We review some of the published work in the following. In general, Fake news could be categorized into three groups. The first group is fake news, which is news that is completely fake and is made up by the writers of the articles. The second group is fake satire news, which is fake news whose main purpose is to provide humor to the readers. The third group is poorly written news articles, which have some degree of real news, but they are not entirely accurate. In short, it is news that uses, for example, quotes from political figures to report a fully fake story. Usually, this kind of news is designed to promote certain agenda or biased opinion [6].

Rubin et al. [7] discuss three types of fake news. Each is a representation of inaccurate or deceptive reporting. Furthermore, the authors weigh the different kinds of fake news and the pros and cons of using different text analytics and predictive modeling methods in detecting them. In this paper, they separated the fake news types into three groups:

- Serious fabrications are news not published in mainstream or participant media, yellow press or tabloids, which as such, will be harder to collect.
- Large-Scale hoaxes are creative and unique and often appear on multiple platforms. The authors argued that it may require methods beyond text analytics to detect this type of fake news.
- Humorous fake news, are intended by their writers to be entertaining, mocking, and even absurd. According to the authors, the nature of the style of this type of fake news could have an adverse effect on the effectiveness of text classification techniques.

The authors argued that the latest advance in natural language processing (NLP) and deception detection could be helpful in detecting deceptive news. However, the lack of available corpora for predictive modeling is an important limiting factor in designing effective models to detect fake news.

Horne et al. [8] illustrated how obvious it is to distinguish between fake and honest articles. According to their observations, fake news titles have fewer stop-words and nouns, while having more nouns and verbs. They extracted different features grouped into three categories as follows:

- Complexity features calculate the complexity and readability of the text.
- Psychology features illustrate and measure the cognitive process and personal concerns underlying the writings, such as the number of emotion words and casual words.
- Stylistic features reflect the style of the writers and syntax of the text, such as the number of verbs and the number of nouns.

The aforementioned features were used to build an SVM classification model. The authors used a dataset consisting of real news from BuzzFeed and other news websites, and Burfoot and Baldwin's satire dataset [9] to test their model. When they compared real news against satire articles (humorous article), they achieved 91% accuracy. However, the accuracy dropped to 71% when predicting fake news against real news.

Wang et al. [10] introduced LIAR, a new dataset that can be used for automatic fake news detection. Thought LIAR is considerably bigger in size, unlike other data sets, this data set does not contain full articles, it contains 12800 manually labeled short statements from politicalFact.com.

Rubin et al. [11] proposed a model to identify satire and humor news articles. They examined and inspected 360 Satirical news articles in mainly four domains, namely, civics, science, business, and what they called "soft news" ('entertainment/gossip articles'). They proposed an SVM classification model using mainly five features developed based on their analysis of the satirical news. The five features are Absurdity, Humor, Grammar, Negative Affect, and Punctuation. Their highest precision of 90% was achieved using only three combinations of features which are Absurdity, Grammar, and Punctuation.

3 Proposed Approach and Models

3.1 N-gram Model

N-gram modeling is a popular feature identification and analysis approach used in language modeling and Natural language processing fields. N-gram is a contiguous sequence of items with length *n*. It could be a sequence of words, bytes, syllables, or characters. The most used n-gram models in text categorization are word-based and character-based n-grams. In this work, we use word-based n-gram to represent the context of the document and generate features to classify the document. We develop a simple n-gram based classifier to differentiate between fake and honest news articles. The idea is to generate various sets of n-gram frequency profiles from the training data to represent fake and truthful news articles. We used several baseline n-gram features based on words and examined the effect of the n-gram length on the accuracy of different classification algorithms.

3.2 Data Pre-processing

Before representing the data using n-gram and vector-based model, the data need to be subjected to certain refinements like stop-word removal, tokenization, a lower casing, sentence segmentation, and punctuation removal. This will help us reduce the size of actual data by removing the irrelevant information that exists in the data.

We created a generic processing function to remove punctuation and non-letter characters for each document; then we lowered the letter case in the document. In addition, an n-gram word based tokenizer was created to slice the text based on the length of n.

Stop Word Removal
Stop words are insignificant words in a language that will create noise when used as features in text classification. These are words commonly used a lot in sentences to help connect thought or to assist in the sentence structure. Articles, prepositions and conjunctions and some pronouns are considered stop words. We removed common words such as, *a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, the, these, this, too, was, what, when, where, who, will,* etc. Those words were removed from each document, and the processed documents were stored and passed on to the next step.

Stemming
After tokenizing the data, the next step is to transform the tokens into a standard form. Stemming simply is changing the words into their original form, and decreasing 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." We use stemming to make classification faster and efficient. Furthermore, we use Porter stemmer, which is the most commonly used stemming algorithms due to its accuracy.

3.3 Features Extraction

One of the challenges of text categorization is learning from high dimensional data. There is a large number of terms, words, and phrases in documents that lead to a high computational burden for the learning process. Furthermore, irrelevant and redundant features can hurt the accuracy and performance of the classifiers. Thus, it is best to perform feature reduction to reduce the text feature size and avoid large feature space dimension. We studied in this research two different features selection methods, namely, Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF). These methods are described in the following.

Term Frequency (TF)
Term Frequency is an approach that utilizes the counts of words appearing in the documents to figure out the similarity between documents. Each document is represented by an equal length vector that contains the words counts. Next, each vector is normalized in a way that the sum of its elements will add to one. Each word count is then converted into the probability of such word existing in the documents. For example, if a word is in a certain document it will be represented as one, and if it is not in the document, it will be set to zero. Thus, each document is represented by groups of words.

TF-IDF
The Term Frequency-Inverted Document Frequency (TF-IDF) is a weighting metric often used in information retrieval and natural language processing. It is a statistical metric used to measure how important a term is to a document in a dataset. A term importance increases with the number of times a word appears in the document, however, this is counteracted by the frequency of the word in the corpus.

One of the main characteristics of IDF is it weights down the term frequency while scaling up the rare ones. For example, words such as “the” and “then” often appear in the text, and if we only use TF, terms such as these will dominate the frequency count. However, using IDF scales down the impact of these terms.

3.4 Classification Process

Figure 1 is a diagrammatic representation of the classification process. It starts with preprocessing the data set, by removing unnecessary characters and words from the data. N-gram features are extracted, and a features matrix is formed representing the documents involved. The last step in the classification process is to train the classifier. We investigated different classifiers to predict the class of the documents. We investigated specifically six different machine learning algorithms, namely, Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), Linear Support Vector Machines (LSVM), K-Nearest Neighbour (KNN) and Decision Trees (DT). We used implementations of these classifiers from the Python Natural Language Toolkit (NLTK).

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

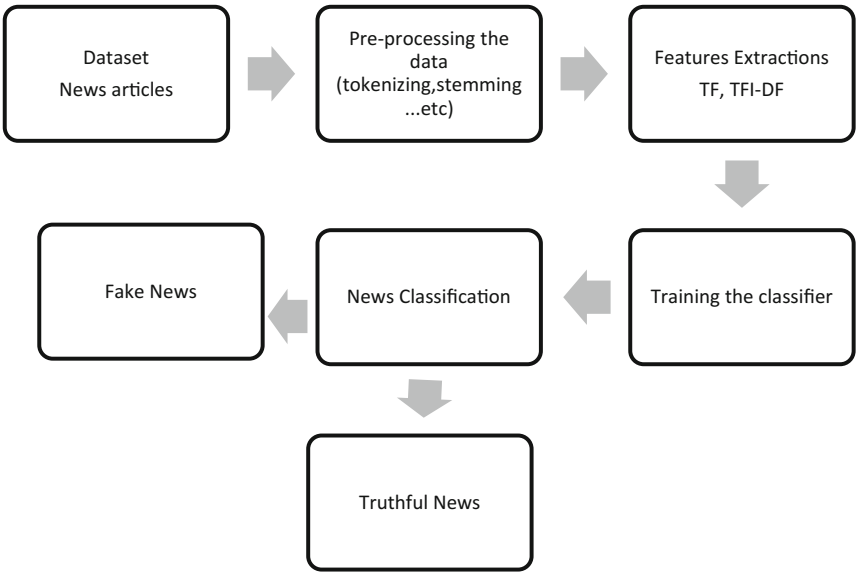

Fig. 1. Classification process

Assume that $\Delta = [d_i]_{1 \le i \le m}$ is our training set consisting of m documents d_i.

Using one of the feature extraction techniques (i.e., TF or TF_IDF), we calculate the feature values corresponding to all the terms/words involved in all the documents in the training corpus and select the p terms $t_j (1 \le j \le p)$ with the highest feature values. Next, we build the features matrix $X = [x_{ij}]_{1 \le i \le m, 1 \le j \le p}$, where:

$$\begin{cases} x_{ij} = feature(t_j) \text{ if } t_j \in d_i \\ x_{ij} = 0 \text{ otherwise} \end{cases}$$

In other words, x_{ij} corresponds to the feature extracted (using TF or TF-IDF) for term t_j for document d_i. Such value is null (0) if the term is not in the document.

4 Experiments

4.1 Dataset

The field of fake news detection is a relatively new area of research. Hence, few public datasets are available. We used in this work primarily a new dataset collected by our team by compiling publicly available news article. We also tested our model on the data set Horne and Adali [8], which is accessible to the public.

Our new dataset was entirely collected from real world sources[1]. We collected news articles from Reuters.com (News website) for real news articles. As for the fake news, they were collected from a fake news dataset on kaggle.com. The collector of the data set collected fake news items from unreliable web sites that Politifact (a fact checking organization in the USA) has been working with Facebook to stamp out. We used 12,600 fake news articles from kaggle.com and, 12,600 truthful articles. We decided to focus only on political news article because these are currently the main target of spammers. The news articles from both fake and truthful categories happened in the same timeline, specifically in 2016. Each of the articles length is bigger than 200 characters.

For every article, the following information is available:

- Article Text
- Article Type
- Article label (fake or truthful)
- Article Title
- Article Date

4.2 Experiments Procedure

We run the aforementioned machine learning algorithms on the dataset, with the goal of predicting whether the articles are truthful or fake. The experiments started by studying the impact of the size (n) of n-grams on the performance. We started with unigram (n = 1), then bigram (n = 2), then steadily increased n by one until reaching n = 4. Furthermore, each n value was tested combined with a different number of features.

The experiments were run using 5-fold cross validation; in each validation round the dataset is divided into 80% for training and 20% for testing.

The algorithms were used to create learning models, and then the learned models were used to predict the labels assigned to the testing data. Experiment results were then presented, analyzed and interpreted. In the start of our research, we applied our model on a combination of news articles from different years with a broader variety of political topics. Our model achieved 98% accuracy when using this type of data. Thus, we decided to collect our data set so we can require fake and real articles from the same year and even same month. Furthermore, we decided to limit the scope of the articles. Thus we only focused on news articles that revolve around the 2016 US elections and the articles that discuss topics around it. In total, we picked 2000 articles from real and fake articles we collected, 1000 fake articles and 1000 real articles. The 2,000 articles represent a subset of the dataset described in the previous section that focuses only politics.

4.3 Experiments Results

We studied two different features extraction methods, TF-IDF, and TF (described earlier), and varied the size of the n-gram from n = 1 to n = 4. We also varied the number of features p (i.e., top features selected), ranging from 1,000 to 50,000. Tables 1, 2, 3, 4, 5 and 6 show the obtained results.

[1] http://www.uvic.ca/engineering/ece/isot/datasets/index.php.

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

N-Gram Size	TF-IDF				TF			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	84.0	86.0	84.0	84.0	85.0	72.0	69.0	69.4
Bi-Gram	78.0	73.0	67.0	54.0	68.0	51.0	47.0	47.0
Tri-Gram	71.0	59.0	53.0	48.0	53.0	47.0	53.0	47.0
Four-Gram	55.0	37.0	37.0	45.0	47.0	48.0	40.0	47.0

Table 2. LSVM Accuracy results. The second row corresponds to the features size. Accuracy values are in %.

N-Gram Size	TF-IDF				TF			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	89.0	89.0	89.0	92.0	87.0	87.0	87.0	87.0
Bi-gram	87.0	87.0	88.0	89.0	86.0	83.0	82.0	82.0
Tri-gram	84.0	85.0	86.0	87.0	86.0	84.0	84.0	79.0
Four-gram	71.0	76.0	76.0	81.0	70.0	70.0	70.0	61.0

Table 3. KNN Accuracy results. The second row corresponds to the features size. Accuracy values are in %.

N-Gram Size	**TF-IDF**				**TF**			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	79.0	83.0	82.0	83.0	77.0	70.0	68.0	68.0
Bi-gram	67.0	65.0	68.0	64.0	62.0	55.0	51.0	45.0
Tri-gram	73.0	68.0	65.0	67.0	76.0	63.0	57.0	46.0
Four-gram	69.0	68.0	68.0	58.0	67.0	54.0	56.0	43.0

Table 4. DT Accuracy Results. The second row corresponds to the features size. Accuracy values are in %.

N-gram Size	**TF-IDF**				**TF**			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	88.0	88.0	89.0	89.0	83.0	88.0	88.0	80.0
Bi-gram	85.0	85.0	85.0	84.0	84.0	87.0	87.0	84.0
Tri-Gram	86.0	86.0	87.0	85.0	86.0	86.0	84.0	86.0
Four-gram	74.0	74.0	71.0	74.0	67.0	67.0	70.0	67.0

Table 5. SGD Accuracy Results. The second row corresponds to the features size. Accuracy values are in %.

N-gram Size	**TF-IDF**				**TF**			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	88.0	86.0	88.0	89.0	87.0	86.0	89.0	85.0
Bi-gram	86.0	85.0	87.0	86.0	85.0	84.0	85.0	84.0
Tri-gram	84.0	85.0	86.0	86.0	85.0	85.0	87.0	87.0
Four-gram	70.0	72.0	74.0	80.0	72.0	73.0	72.0	78.0

Table 6. LR Accuracy Results. The second row corresponds to the features size. Accuracy values are in %.

N-Gram Size	**TF-IDF**				**TF**			
	1000	5000	10,000	50,000	1000	5000	10,000	50,000
Uni-gram	83.0	89.0	89.0	89.0	89.0	89.0	83.0	89.0
Bi-gram	87.0	87.0	88.0	88.0	87.0	85.0	86.0	86.0
Tri-gram	86.0	85.0	88.0	87.0	83.0	83.0	83.0	82.0
Four-gram	70.0	76.0	75.0	81.0	68.0	67.0	67.0	61.0

From the results obtained in our experiments, Linear-based classifiers (Linear SVM, SDG, and Logistic regression) achieved better results than nonlinear ones. However, nonlinear classifiers achieved good results too; DT achieved 89% accuracy. The highest accuracy was achieved using Linear SVM as 92%. This classifier performs well no matter the number of feature values used. Also with the increase of n-gram (Tri-gram, Four-gram), the accuracy of the algorithm decreases. Furthermore, TF-IDF outperformed TF. The lowest accuracy of 47.2% was achieved using KNN and SVM with four-gram words and 50,000 and 10,000 feature values.

We conducted additional experiments by running our model on the dataset of Adali and Horne [8], consisting of real news from BuzzFeed and other news websites, and satires from Burfoot and Baldwin's satire dataset. We obtained 87% accuracy using n-gram features and LSVM algorithm when classifying fake news against real new, which is much better than the 71% accuracy achieved by the authors on the same dataset.

5 Conclusion

The problem of fake news has gained attention in 2016, especially in the aftermath of the last US presidential elections. Recent statistics and research show that 62% of US adults get news on social media [12, 13]. Most of the popular fake news stories were more widely shared on Facebook than the most popular mainstream news stories [14]. A sizable number of people who read fake news stories have reported that they believe them more than news from mainstream media. Dewey [15] claimed that fake news played a huge role in the 2016 US election and that they continue to affect people opinions and decisions.

In this paper, we have presented a detection model for fake news using n-gram analysis through the lenses of different features extraction techniques. Furthermore, we investigated two different features extraction techniques and six different machine learning techniques. The proposed model achieves its highest accuracy when using unigram features and Linear SVM classifier. The highest accuracy score is 92%.

Fake news detection is an emerging research area with few public datasets. We run our model on an existing dataset, showing that our model outperforms the original approach published by the authors of the dataset. In our future work, we will run our model on the few other publicly available datasets, such as the LIAR dataset which was released only recently, after we completed the current phase of our research [10].

References

1. The Verge: Your short attention span could help fake news spread (2017). https://www.theverge.com/2017/6/26/15875488/fake-news-viral-hoaxes-bots-information-overload-twitter-facebook-social-media. Accessed 16 Aug 2017
2. Lemann, N.: Solving the Problem of Fake News. The New Yorker (2017). http://www.newyorker.com/news/news-desk/solving-the-problem-of-fake-news

3. Schulten, K.: Skills and Strategies—Fake News vs. Real News: Determining the Reliability of Sources. The Learning Network (2017). https://learning.blogs.nytimes.com/2015/10/02/skills-and-strategies-fake-news-vs-real-news-determining-the-reliability-of-sources/. Accessed 16 Aug 2017
4. Levin, S.: Pay to sway: report reveals how easy it is to manipulate elections with fake news. The Guardian (2017). https://www.theguardian.com/media/2017/jun/13/fake-news-manipulate-elections-paid-propaganda
5. Gu, L., Kropotov, V., Yarochkin, F.: The fake news machine, how propagandists abuse the internet and manipulate the public. In: 1st ed. [pdf] Trend Micro, p. 81 (2017). https://documents.trendmicro.com/assets/white_papers/wp-fake-news-machine-how-propagandists-abuse-the-internet.pdf?_ga=2.117063430.1073547711.1497355570–1028938869.1495462143
6. Schow, A.: The 4 Types of 'Fake News'. Observer (2017). http://observer.com/2017/01/fake-news-russia-hacking-clinton-loss/
7. Rubin, V.L., Chen, Y., Conroy, N.J.: Deception detection for news: three types of fakes. In: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (ASIST 2015). Article 83, p. 4, American Society for Information Science, Silver Springs (2015)
8. 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)
9. Burfoot, 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)
10. Wang, W.Y.: Liar, Liar Pants on fire: a new Benchmark dataset for fake news detection. arXiv preprint (2017). arXiv:1705.00648
11. Rubin., Victoria, L., et al.: Fake news or truth? Using satirical cues to detect potentially misleading news. In: Proceedings of NAACL-HLT (2016)
12. Gottfried, J., Shearer, E.: News use across social media platforms. Pew Res. Cent. 26 (2016)
13. Gottfried, J., et al.: The 2016 presidential campaign–a news event that's hard to miss. Pew Res. Cent. 4 (2016)
14. Silverman, C., Singer-Vine, J.: Most americans who see fake news believe it, new survey says. BuzzFeed News (2016)
15. Dewey, C.: Facebook has repeatedly trended fake news since firing its human editors. Washington Post (2016)