Content Based Fake News Detection Using N-Gram Models

Hnin Ei Wynne†
Department of Computer Engineering and Information Technology
Mandalay Technological University,Myanmar
hnineiwynne@mtu.edu.mm

Zar Zar Wint
Department of Computer Engineering and Information Technology
Mandalay Technological University,Myanmar
zarzarwint@mtu.edu.mm

ABSTRACT

Fake news is very popular these days because of the increasing popularity of social media. Detecting fake news is considered as one of the most dangerous types of deception because it is created with dishonest intention to misdirect the public. Many researchers proposed fake news detection systems considering many approaches; content, social-context, and propagation. When the news is detected fake or real, there is a limitation in the accuracy and understandability of language. In this paper, we propose the fake news detection system that considers the content of the online news articles. We investigate two machine learning algorithms with the use of word n-grams and character n-grams analysis. Experiments yield better results using character n-grams with Term-Frequency-Inverted Document Frequency (TF-IDF) and Gradient Boosting Classifier achieves an accuracy of 96%.

CCS CONCEPTS

• Information systems • Computing methodologies→ Lexical semantics

KEYWORDS

Online fake news, Fake news detection, Word n-gram, Character n-grams

ACM Reference format:

Hnin Ei Wynne and Zar Zar Wint. 2019. Content Based Fake News Detection Using N-gram Models. In *Proceedings of the 21st International conference on Information Integration and Web-based Applications & Services (iiWAS2019). ACM, Munich, Germany, 5 pages.*
https://doi.org/10/1145/3366030.3366116

iiWAS2019, December 2-4, 2019, Munich, Germany

ACM ISBN 978-1-4503-7179-7/19/12$15.00
https://doi.org/10/1145/3366030.3366116

1 Introduction

Nowadays social media are growing in popularity. Most of the people consume news from social media rather than traditional news media because of timely, low cost and easy access. In recent years, fake news has increasingly become a dangerous prospect for online users. The term “fake news” becomes popular because of spreading the various kinds of fake news during the 2016 US presidential elections [1].

One of the challenges in detecting fake news is that there does not yet exist a proper definition of fake news and the acute characterization of fake news required to label an article as true or false [2]. Fake news is defined as the prediction of the chances of a particular news article (news report, editorial, expose, etc.) being intentionally deceptive [3]. In the Cambridge dictionary, fake news is defined that false stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke.

Along the news is disseminated very fast in the social networks, the fake news is spread automatically. The ever-growing influence of fake news is felt daily from politics to education and financial markets, as it is difficult to identify fake news and genuine news. Besides, the language used in fake news and real ones are very similar because fake news is created with the intention to be trusted. Therefore, fake news detection has become a very important task, but yet technically very challenging.

In general, current content-based approaches detect fake news by representing news content in terms of features within a machine learning framework and focus mainly on the linguistic indicators of fake news. Moreover, deep neural networks, which have been successful in natural language processing, recently used to detect automatically [4, 5, 6, 7, 8].

Riedel et al. [9] show that using a relatively simple approach based on term frequency (TF) and term frequency-inverse document frequency (TF-IDF) with a multilayer perceptron has obtained a good accuracy of 88.5% in the context of fake news challenge dataset. Ahmed [10] yields an accuracy of 92% using TF-IDF and Linear Support Vector Machine (LSVM) with unigram on a 2000 news dataset. Bharadwaj [11] has recently used TF, TFIDF with n-grams features, Random Forest with bigram features achieve the best accuracy of 95.66%.

Some researchers used a combination of n-grams and syntax information. Mukherjee et al. [12] have used the bi-grams with part-of-speech (POS) tagging to achieve a 68.1% accuracy on Yelp data. It is considered that the accuracy of the different approaches depends largely on the dataset, which is being used. Rubin [2] has noted that simple content-related n-grams and part-of-speech (POS) tagging have been proven insufficient for the classification task. Rather, they suggested Deep Syntax analysis using Probabilistic Context-Free Grammars (PCFG). Gilda [13] applies the TF-IDF of bigrams and Probabilistic Context-Free Grammar (PCFG) detection to 11000 corpora. They report that they have achieved an accuracy of 77.2% for the Stochastic Gradient Descent model when bi-gram TF-IDF was used. However, they did not experiment with different lengths of n-grams (uni-gram, tri-gram, four-gram) and bigram TF-IDF performs well for their modified dataset. Their model depends largely on the dataset.

Several studies in detecting fake and real news have focused only on the word n-grams features in their work. In our study, we consider applying both word n-grams and character n-grams to the fake news detection system. TF and TF-IDF are also used as feature extraction techniques. Our objective is to analyze word n-grams and character n-grams features on two machine learning algorithms. It is intended to understand what features are most predictive of fake news.

The rest of this paper is prepared as follows. Section 2 overviews the related work. Section 3 shows the method. Section 4 shows the evaluation and results. Finally, the conclusion of this paper is described in Section 5.

2 Related Work

The problem of fake news detection has become an emerging topic in recent social media studies. Several solutions are proposed for this problem. Fake news detection can be considered as text classification issues. Prior work has shown that the use of n-grams as a feature in fake news detection problem.

Ahmed et al. [10] presented a detection model for fake news using six machine learning algorithms with n-gram features in their work. They used TF and TF-IDF for feature extraction and analyzed different sizes of the word n-grams in their model. Their model achieves the highest accuracy when using unigram features and a linear SVM classifier. They showed the effectiveness of using word n-grams in fake news detection. In contrast, we analyzed character n-grams besides the word n-grams to know which features could predict better.

Bharadwaj [11] applied semantic features such as unigram TF, unigram TF-IDF, bi-gram, tri-gram, quad-gram, and GloVe word embedding along with Naïve Bays, Random Forest, and RNN classifiers to detect fake news. Bi-gram outperforms other n-grams features.. They considered the word n-grams features for detecting fake news. They concluded that word pairs best indicate the authenticity of the news In our research work, the various lengths of n-grams are considered for both TF and TF-IDF.

Gilda [13] showed that while bi-gram TF-IDF yields highly effective models for detecting fake news, the PCFG features do little efficient. They examined their dataset over more than one class algorithms to find out the great model. They presented that TF-IDF of bi-grams fed right into a Stochastic Gradient Descent model with an accuracy of 77.2%. They do not consider the different lengths of word n-grams (uni-gram, bi-gram, tri-gram, etc.). They concluded that their word bi-gram TF-IDF does well in their modified dataset. Their model would not be robust when changing news cycles, which may require a more complete corpus. In our studies, we investigate the different lengths of word n-gram with TF-IDF to know which features pair best for fake news detection. We also analyzed the n-grams on two different datasets to know that the models can give good results on different datasets.

Most of the prior work for fake news detection focused on word n-gram features. However, A.Hafiz [14] presented the feasibility of combining word and character n-grams as a feature for detecting deceptive opinions using standard hotel reviews dataset. In [15], the authors explored the use of word and character n-grams combinations, function words and the word syntactic n-grams as features for classifiers in detecting deceptive and true opinion. They showed the robustness of word and character n-grams combination for deception related classification. We have been motivated me to explore the use of character n-grams features in fake news detection problem.

In this research, our approach is to analyze word n-grams and character n-grams features for detecting fake news. Moreover, the two machine learning algorithms predict fake news by using different lengths of n-grams (uni-gram, bi-gram, tri-gram, four-gram) with TF and TF-IDF. We intend to examine the robustness of our features by utilizing two datasets.

3 Method

In this section, we describe our general approach to detect fake news using the word and character n-grams features.

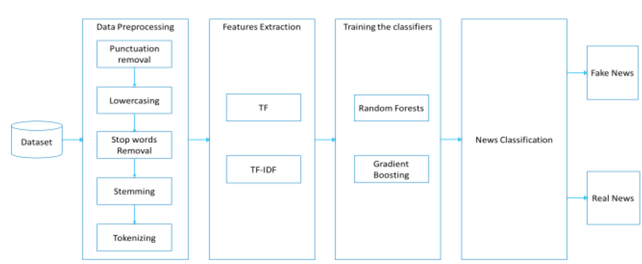


Figure 1: The general overview of our model

Figure 1 shows a general overview of our proposed model. The first step is preprocessing the dataset by cleaning the unstructured data and the word and character n-gram features are extracted. N-grams features are extracted and a feature matrix is formed to represent the content. We use 10-fold cross-validation and so the dataset is split into 90% training data and 10% testing data. TF and TF-IDF are feature extraction techniques and calculate the feature values for the corresponding to all words in all contents in the training set. The next step is training the classifiers. We utilize two machine learning algorithms, namely, Random Forest and Gradient Boosting. These machine learning algorithms learn from the provided training data and classify the news into fake or real.

3.1 Data Pre-processing

We applied the data preprocessing steps to both the headlines and the news articles to reduce the size of the actual data. Raw texts of news are an unstructured form of data and could contain noisy content. The raw texts need to be preprocessed before extracting the features for the models. There are different ways to convert the text into a form that is ready for modeling.

Firstly, we eliminated the punctuation. Next, we transformed the capital-letter information into the lower cases in the document. We removed the stop words that are insignificant in a language and create noise when used as features in text classification. The next step is stemming that change the words into their original form. Then, we tokenize the title and body of each article based on lengths of n-grams.

3.2 N-grams

N-grams are contiguous sequences of n-items in a text or speech. The n refers to the number of combinations of items, which can be phonemes, syllables, letters, words, character, byte or any sequence of data. The wider used of n-grams model in natural language processing are work-based n-gram and character-based n-grams.

In this research work, we use both word and character n-grams as features. The size of the n-gram can have different naming conventions. An n-gram of one word or character is called uni-gram, two is bi-gram, three is tri-gram and four-gram is a term where n=4.

For example, given the sentence "Achieving starts with believing", we can extract the word n-gram as shown below:

Uni-grams: Achieving, starts, with, believing.
Bi-grams: Achieving starts, starts with, with believing
Tri-gram: Achieving starts with, starts with believing
Four-gram: Achieving starts with believing

As an example, the word "Thanks" is tokenized based on a character level is the following:

Uni-grams: T, h, a, n, k, s
Bi-grams: Th, ha, an, nk, ks
Tri-grams: Tha, han, ank, nks
Four-grams: Than, hank, anks

In this research, we use word and character n-grams to represent the content of the news and the various sets of n-gram frequency profiles are generated from the training data to represent real and fake news.

3.3 Feature Extraction Techniques

We used two feature extraction techniques, namely Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF).

3.3.1 Term Frequency (TF). TF is the number of occurrences of a term in a document in the corpus. It is the ratio of the number of times the word appears in a document compared to the total number of words in that document. It increases as the number of occurrences of that word within the document increases. Each document has its own tf. Equation (1) shows how to find the term frequency.

$$TF(i,j) = \frac{Term\ i\ frequency\ in\ document\ j}{Total\ words\ in\ document\ j} \qquad (1)$$

*3.3.2 Inverse Document Frequency (ID*F). IDF measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and " that", may appear a lot of times but have little importance. Thus, we need to calculate the weight of rare words across all documents in the corpus. The words that occur rarely in the corpus have a high IDF score. It is given by Equation (2).

$$IDF(i) = \log(\frac{Total\ Documents}{documents\ with\ term\ i}) \qquad (2)$$

3.3.3. Term Frequency- Inverted Document Frequency (TF-IDF). Combining TF and IDF, we come up with the TF-IDF score for a word in a document in the corpus. It is the product of tf and idf.

TF-IDF score for term i in document j =TF(i,j)* IDF(i) (3)

We used TfidfVectorizer function of python sklearn.feature_extraction library to generate TF-IDF n-gram features in this research work.

4 Evaluation

4.1 Dataset

We utilized two kaggle datasets (real_or_fake news dataset and fake news detection dataset) to test whether the n-grams model allows addressing the detection of fake news. Dataset 1 (real or fake news dataset) contains 6256 articles including their title. Half of them are labeled as real and the other half are fake. Each article contains the title, text, and label. Dataset 2 (fake news detection dataset) contains 4009 articles. The corpus has 2137 deceptive news and 1872 truth news. Each article consists of URLS, headline, body, and label.

4.2 Experiment Settings

The purpose of this work is to analyze the word n-grams and the character n-grams features for detecting fake news. We did some experimental settings. At the preprocessing step, we utilized Natural Language Toolkit (NLTK) converting to lowercase, removing punctuation, removing stop words and tokenization. For representing n-grams features, we used TF and TF-IDF as feature extraction techniques and we use scikit-learn to calculate the TF-IDF for each n-gram and build a sparse matrix of the resulting features. For classification, we applied Random Forest and Gradient Boosting in this study. All experiments were run using 10-fold cross-validation. We split the dataset into 90% for training and 10% for testing.

4.3 Experiment 1

In this experiment, we used the Dataset 1, intending to predict whether the news is real or fake. We varied the size of the n-gram from n=1 to n=4. We also varied the number of features, ranging from 1000 to 10000. We used accuracy as the metric for evaluating fake news detection problem. Table 1 shows the accuracy evaluation of fake news detection model by comparing two n-grams features (character n-grams and word n-grams) through the lengths of n-grams

Table 1. Results using character n-grams and word n-grams as features on Dataset 1. Results in bold indicate the best-obtained detection results for real_or_fake news dataset.

Feature extraction techniques	Classifiers	N-gram models	Lengths of n-gram			
			Uni-gram	Bi-gram	Tri-gram	Four-gram
TF (1000 features)	Random Forests	char	0.87	0.89	0.83	0.81
		word	0.86	0.84	0.78	0.77
	Gradient Boosting	char	0.89	0.94	0.88	0.87
		word	0.89	0.84	0.79	0.73
TF-IDF (1000 features)	Random Forests	char	0.85	0.89	0.83	0.82
		word	0.86	0.83	0.81	0.77
	Gradient Boosting	char	0.89	**0.95**	0.89	0.88
		word	0.91	0.85	0.82	0.73
TF (5000 features)	Random Forests	char	0.86	0.89	0.89	0.86
		word	0.86	0.85	0.81	0.79
	Gradient Boosting	char	0.89	0.94	0.88	0.87
		word	0.89	0.84	0.79	0.73
TF-IDF (5000 features)	Random Forests	char	0.86	0.87	0.88	0.87
		word	0.86	0.87	0.88	0.79
	Gradient Boosting	char	0.88	0.94	**0.95**	0.94
		word	0.92	0.86	0.80	0.74
TF (10000 features)	Random Forests	char	0.86	0.87	0.89	0.84
		word	0.83	0.83	0.84	0.80
	Gradient Boosting	char	0.89	0.94	0.94	0.93
		word	0.89	0.86	0.72	0.72
TF-IDF (10000 features)	Random Forests	char	0.86	0.87	0.89	0.87
		word	0.84	0.84	0.84	0.81
	Gradient Boosting	char	0.89	0.94	**0.95**	0.94
		word	0.89	0.86	0.80	0.74

From the results obtained in experiment 1, the highest accuracy 95% was achieved using Gradient Boosting with character n-grams (bi-gram and tri-gram). Through the various lengths of n-grams, the classifiers using character n-grams as features achieved better results than the classifiers using word n-grams.

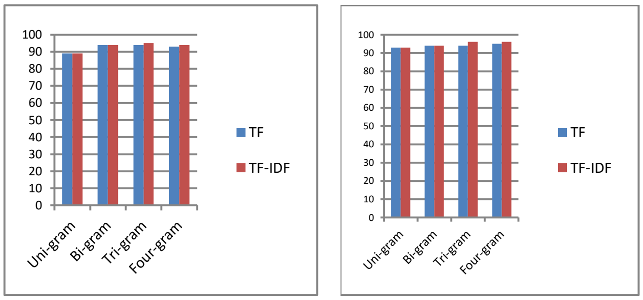
4.4 Experiment 2

In this experiment, we used the Dataset 2 (fake news detection dataset), intending to predict whether the news is real or fake. We varied the size of the n-gram from n=1 to n=4 and the number of features, ranging from 1000 to 10000 like experiment 1. Table 2 shows the comparison of fake news detection models in terms of accuracy.

Table 2. Results using character n-grams and word n-grams as features on Dataset 1. Results in bold indicate the best-obtained detection results for fake news detection dataset.

Feature extraction techniques	Classifiers	N-gram models	Lengths of n-gram			
			Uni-gram	Bi-gram	Tri-gram	Four-gram
TF (1000 features)	Random Forests	char	0.92	0.92	0.89	0.89
		word	0.91	0.90	0.91	0.89
	Gradient Boosting	char	0.92	0.94	0.93	0.92
		word	0.91	0.92	0.91	0.87
TF-IDF (1000 features)	Random Forests	char	0.91	0.92	0.88	0.88
		word	0.91	0.91	0.92	0.91
	Gradient Boosting	char	0.92	0.94	0.93	0.92
		word	0.91	0.92	0.91	0.87
TF (5000 features)	Random Forests	char	0.91	0.92	0.92	0.91
		word	0.91	0.92	0.92	0.91
	Gradient Boosting	char	0.92	0.93	0.93	0.94
		word	0.94	0.92	0.91	0.87
TF-IDF (5000 features)	Random Forests	char	0.91	0.92	0.92	0.93
		word	0.91	0.92	0.92	0.92
	Gradient Boosting	char	0.92	0.93	0.94	0.94
		word	0.94	0.93	0.90	0.90
TF (10000 features)	Random Forests	char	0.92	0.93	0.93	0.93
		word	0.92	0.94	0.94	0.92
	Gradient Boosting	char	0.93	0.94	0.94	0.95
		word	0.94	0.94	0.93	0.88
TF-IDF (10000 features)	Random Forests	char	0.93	0.93	0.95	0.94
		word	0.92	0.94	0.94	0.93
	Gradient Boosting	char	0.93	0.94	**0.96**	**0.96**
		word	0.94	0.94	0.92	0.91

From the experiment 2, Both Gradient Boosting and Random Forests achieved good results. However, Gradient Boosting achieved the highest accuracy of 96% when using character tri-gram and character four-gram TF-IDF at 10,000 features.



(a) (b)

Figure 2: Gradient Boosting accuracy values using character n-grams with 10000 features (a) in experiment 1 and (b) in experiment 2

Moreover, character n-grams performed better than word n-grams in both classifiers. From the results obtained from both experiments, using the character n-grams achieved better accuracy than using word n-grams in both classifiers.

Moreover, TF-IDF outperforms TF when the number of features values used is 10000. Especially the size of the n-gram is 3 and 4. Figure 2 shows Gradient Boosting accuracy comparison of TF and TF-IDF in experiment 1 and experiment 2. The figure represents only using the character n-grams. However, TF-IDF achieved better results than TF when using both character n-grams and word n-grams with 10000 features. Furthermore, when the size of word n-grams increases the accuracy of classifiers decrease.

5 Conclusion

Fake news detection becomes a critical challenging problem nowadays. Many research works have been done upon news content and social context to detect fake news. Most of the studies focus on word n-gram as a feature while detecting fake news. In this study, we have addressed the fake news detection problem by using both word n-grams and character n-grams. Even though character n-gram has been used in many classification tasks, to the best of our knowledge, this is the first time using character n-grams in detecting fake news. Our experimental results show that character n-grams outperform work n-grams using TF and TF-IDF as feature extraction techniques. Interestingly, we noted that the length of the word n-grams increased, the accuracy of the classifiers decreased. In our going and future work, we will consider some possible combination of n-grams and also consider social context features besides content to improve the classification results in the fake news detection.

.

ACKNOWLEDGMENTS

This research has been funded by the Japan International Cooperation Agency (JICA) EEHE project. The authors acknowledge Professor Masayoshi Aritsugi from Kumamoto University (Japan) for his valuable comments that significantly improved this work.

REFERENCES

[1] Hunt Allcott and Matthew Gentzkow. 2017. Social Media and Fake News in the 2016 Election. Journal of Economic Perspectives 31, 2 (2017), 211-236.

[2] David M. J. Lazer, Matthew Baum, Yochai Benker, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, C. Sunstein, Emily A. Thorson, Duncan J. Watts and Jonthan L. Zittrain. 2018. The science of fake news. Science, 359(6380): 1094-1096.

[3] Niall J. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. In Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community,51,1(2015),St. Louis, MO, USA, 1-4.

[4] Mykhailo Granik and Volodymyr Mesyura. 2017. Fake News Detection Using Naïve Bays Classifier. 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). IEEE, 900-903.

[5] Supanya Aphiwongsophon and Prabhas Chongstitvatana. 2018. Detecting Fake News with Machine Learning Method. 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Chiang Rai, Thailand, 528-531.

[6] Victoria L Rubin, Nadia Conroy, and Yimin Chen. 2015. Towards news vertification: Deception detection methods for news discourse. In Proceedings of the Hawaii Internatonal Conference on System Sciences (HICSS48) Symposium on Rapid Screening Technologies, Deception Detection and Credibility Assessment Symposium, 5-8.

[7] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2018. Fake News Identification on Twitter with Hybrid CNN and RNN Models. In Proceedings of the International Conference on Social Media & Sociery, Copenhagen, Denmark (SMSociety). ACM Press, Cophagen, Denmark. DOI: https// doi.org/10.1145/3217804.3217917

[8] Hamid Karimi, Proteek Chandan Roy, Sari Saba-Sadiya, and Jiliang Tang. 2018. Multi-Source Multi-Class Fake News Detection. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, 20-26.

[9] Benjamin Riedel, Isabelle Augenstein, Georgios P.Spithourakis, and Sebastian Riedl. 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task, arXiv: 1707.03264[cs]

[10] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques, in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, ser. Lecture Notes in Computer Science. Springer, Cham, 127-138.

[11] Pranav Bharadwaj and Zongru Shao. 2019. Fake News Detection with Semantic Features and Text Mining. hal-02071d914.

[12] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. What yelp fake review filter might be doing? In Procedings of the 7th International AAAI Conference on Weblogs and Social Media, 409-418.

[13] Shlok Gilda. 2017. Evaluating Machine Learning Algorithmsfor Fake News Detection, 2017 IEEE 15th Student Conference on Research and Development (SCOReD), 110-115.

[14] Al Hafiz Kabar Maulana Siagian and Masayoshi Aritsugi. 2017. Combining Word and Character N-Grams for Detecting Deceptive Opinions. In 2017 IEEE 41 Annual Computer Software and Applications Conference (COMPSAC). IEEE Computer Society, Washington DC, USA.

[15] Al Hafiz Kabar Maulana Siagian and Masayoshi Aritsugi (2019). Robustness of Word and Character N-grams Combinations in Detecting Deceptive and Truthful Opinions, ACM Journal of Data and Information Quality. DOI: https://jdiq.acm.org/index.cfm#