Received: 3 September 2017 | Revised: 5 November 2017 | Accepted: 12 November 2017 | Published on: 29 December 2017

DOI: 10.1002/spy2.9

ORIGINAL ARTICLE

WILEY

Detecting opinion spams and fake news using text classification

Hadeer Ahmed[1] | **Issa Traore**[1] | **Sherif Saad**[2]

[1]ECE Department, University of Victoria, Victoria, Canada
[2]School of Computer Science, University of Windsor, Windsor, Canada

Correspondence
Traore, Issa, ECE Department, University of Victoria, Victoria, BC, Canada.
Email: itraore@ece.uvic.ca

In recent years, deceptive content such as fake news and fake reviews, also known as opinion spams, have increasingly become a dangerous prospect for online users. Fake reviews have affected consumers and stores alike. Furthermore, the problem of fake news has gained attention in 2016, especially in the aftermath of the last U.S. presidential elections. Fake reviews and fake news are a closely related phenomenon as both consist of writing and spreading false information or beliefs. The opinion spam problem was formulated for the first time a few years ago, but it has quickly become a growing research area due to the abundance of user-generated content. It is now easy for anyone to either write fake reviews or write fake news on the web. The biggest challenge is the lack of an efficient way to tell the difference between a real review and a fake one; even humans are often unable to tell the difference. In this paper, we introduce a new n-gram model to detect automatically fake contents with a particular focus on fake reviews and fake news. We study and compare 2 different features extraction techniques and 6 machine learning classification techniques. Experimental evaluation using existing public datasets and a newly introduced fake news dataset indicate very encouraging and improved performances compared to the state-of-the-art methods.

KEYWORDS

fake content detection, online fake news, online fake reviews, online social network security, opinion spams, text classification

1 | INTRODUCTION

Opinions such as online reviews are the main source of information for customers to help gain insight into the products they are planning to buy.[1] Customers write reviews to provide feedback by sharing their experience, either bad or good, with others. Their experiences impact businesses for the long term either positively or negatively. Naturally, this creates incentives and opportunities for manipulating customers' decisions by generating false/fake reviews.[2–7] Such a practice is called opinion spamming where spammers will write false opinions to influence others. Opinions/Reviews may be produced either to improve or damage the reputation of a business/product.

Fake reviews can broadly be categorized into 3 groups.[8] First, there are untruthful reviews, whose main purpose is to present false information about the product either to enhance its reputation or to damage it. Second are reviews that target the brand but do not express an experience with a certain product. The third group is nonreviews and advertisements that contain text only indirectly related to the product. Groups 3 and 2 are relatively easy to identify, while the first one is a bit problematic. These reviews are either written by a single spammer hired by the business owner or a group of spammers who work together in a time window to manipulate the reputation of a product or store.

Recently, it has become apparent that opinion spam does not only exist in product reviews and customer feedback. In fact, fake news and misleading articles is another type of opinion spam.[9,10] One of the biggest sources of spreading fake news or rumors is, of course, social media websites,[11] such as Google Plus, Facebook, Twitter, and so on. Like fake reviews, fake news can also be categorized broadly into 3 groups. The first is false 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

 Copyright © 2017 John Wiley & Sons, Ltd.

to the readers. The third group is poorly written news articles, which have a certain degree of real news but are not entirely accurate. In short, it is news that uses, for example, quotes from political figures to report an entirely fake story. Usually, this kind of news is designed to promote a certain agenda or biased opinion.[12]

Detecting fake news is believed to be a complex task and much harder than detecting fake product reviews.[13] The open nature of the web and social media, in addition to the recent advance in computer technologies, simplifies 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, customers, 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 requires measuring the news propagation, which has shown to be complex and resource intensive.[14,15]

In our study, we explore the 2 main classes of opinion spam, namely, fake reviews and fake news. The commonality between the different classes of opinion spam is that they are all fake content. Hence, we introduce a unified approach for the automatic detection of fake content, applicable for detecting both fake reviews and news.

We propose a detection model that combines text analysis using n-gram features and terms frequency metrics and machine learning classification. We investigate 2 different features extraction techniques and 6 different machine learning classification techniques. The proposed model is evaluated using 3 different datasets, involving truthful and fake content (both news articles and reviews). Experimental evaluation indicates that our model outperforms existing work using related approaches. We achieved an accuracy of 90% on Ott et al.'s dataset,[16] which is slightly higher than the 89% achieved by Ott et al. on the same dataset. Using the dataset of Horne and Adali,[17] we achieved an accuracy of 87%, which is considerably higher than the accuracy of 71% obtained by Horne and Adali using text features.

Although we used some existing public datasets, we also collected a new dataset for the study of fake news, which contains 12 600 fake news articles and 12 600 legitimate news article. The new dataset will represent a useful corpus for researchers working in the emerging field of fake news identification. By running our model on the aforementioned news dataset, we obtain an accuracy of 92%, which is very encouraging.

The rest of the paper is organized as follows. Section 2 is a review of related works. Section 3 introduces our fake content detection model. Section 4 presents 2 experiments conducted to evaluate the accuracy of our model in detecting fake reviews and fake news, respectively. Section 5 concludes the paper by summarizing our contributions and outlining future work.

2 | REVIEW OF RELATED LITERATURE

In this section, we review existing works related to opinion spam detection and the various methods used to detect fake reviews.

2.1 | Opinion spam detection

The opinion spam detection problem was first introduced by Jindal et al.[8] in 2008. They categorized reviews broadly into 3 types: fake reviews, reviews targeting an individual brand, and reviews advertising a product. They analyzed 10 million reviews from Amazon.com to showcase and detect fake reviews. Identifying the last 2 categories (type 2 and type 3) was made easy by simply using standard supervised machine learning classifiers such as naïve Bayes (NB) and logistic regression (LR). They also constructed 36 features based on review content, reviewer behavior, and product description and statistics. However, detecting type 1 (fake reviews) was tricky as there was no available labeled data. Thus, they labeled duplicated and near-duplicated reviews as fake opinions and the remaining data as truthful opinions. They used LR to build their model, which has the added benefit of producing a probability estimating the likelihood of a review to be fake. Besides the LR, they also tried support vector machine (SVM), NB, and decision tree. They were able to identify type 2 and type 3 with 98.7% accuracy. As for type 1, the accuracy was 78% using all features and 63% using text features only. The results also showcase that only reviews and reviews by top rank reviewers are most likely spam. Furthermore, products generating high sales numbers will receive less spam. They were able to uncover certain information about the relationship between reviews, reviewers, products, ratings, and feedback on reviews. Normally, reviewers do not write a large number of reviews; an individual product does not get a significant amount of reviews, and reviews do not get much feedback.

Most models designed to detect opinion spam (ie, fake reviews) tend to follow 2 directions. The first is to engineer the features from review content; the second uses features engineered from the spammer behavior.

2.2 | Content-based detection models

One of the main review content-based detection models was developed by Ott et al.,[16] who used n-gram term frequency to detect fake opinions. They created a "gold-standard" dataset by collecting deceptive opinions of hotels from Amazon Mechanical

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Turk and honest opinions from TripAdvisor. They divided all the opinions (fake and truthful) into positive and negative groups. Using an SVM classifier, they achieved 86% accuracy. When they removed the positive and negative separation, the accuracy of the model dropped from 86% to 84%, which implied that separating the data into negative and positive groups improves the performance. Furthermore, they established the inability of humans to identify fake reviews efficiently. They employed humans to judge the reviews. The highest score for a human judge was 65%.

Another content-based detection model was developed by Mukherjee et al.,[18] who contended the validity of using pseudo fake reviews such as Ott et al's[16] gold-standard dataset to detect fake reviews. Ott et al.[16] were able to achieve 89.6% accuracy with only n-gram features. Mukherjee et al. argued that pseudo reviews, which are generated reviews, do not count as real fake reviews as these do not represent (unsolicited or spontaneous) real-world fake reviews. They decided to test their methods on filtered reviews from Yelp as these will be more trustworthy. They were able to achieve only 67.8% accuracy when they tested Ott et al's[16] model on Yelp data. Thus, they believed results from models trained using pseudo fake reviews are not accurate, and the methods are useless when used on real-world data. However, they acknowledged that 67.8% is still impressive. Thus, n-gram is still useful in detecting fake reviews. Also, they proposed a method to figure out the difference between 2 types of review data. In a later study, Mukherjee et al.[19] claimed that Yelp spammers seem to exaggerate faking by using words and phrases that appear genuine and attempt not to reuse certain words a lot.

Some of the content-based detection models, such as the work by Shojaee et al., rely on stylometry analysis. Shojaee et al.[20] developed a stylometric-based model for review classification. The researchers used the "gold-standard" dataset created by Ott et al.[16] They extracted 234 stylometric features, divided them into lexical and syntactic features, and used SVM and NB models for classification. First, they tested the lexical and syntactic features separately and then a combination of both by computing the F-measure. SVM outperformed NB with the features combined or separated. However, the highest F-measure score was 84% using both lexical and syntactic features.

Other content-based detection models involve the use of semantic similarity metrics, such as the proposals of Algur et al.[21] and Lau et al.[22]

Algur et al.[21] proposed a conceptual-level similarity model to detect fake reviews. The proposed approach handles similar words by mapping the words to abstract concepts. They focused mainly on product features, mentioned by reviewers in their reviews, to calculate the similarities between reviews. According to the model evaluation results, conceptual-level similarity scored lower than human annotation when detecting spam reviews. While human annotation of reviews, in terms of pros and cons, achieved accuracies of 92.91% and 94.37%, respectively, the conceptual model scored 57.29% and 30.00%, respectively. Although the proposed approach was able to detect a significant amount of spam reviews, its results in comparison with human perception indicate that it is not well suited to detect spam in general.

Lau et al.[22] used unsupervised techniques to address the unavailability of labeled data and situations where prominent features are not available. They created a method to calculate the fakeness of a review by estimating the semantic overlap of content between the reviews. The model built on the idea that a spammer is a human in the end, and humans lack imagination, which means they will tend to reuse their reviews. They will attempt to make it appear more honest by changing words; for example, they will change "love" to "like." Their research was the first to apply inferential language modeling and text mining to detect fake reviews. They compiled their dataset by gathering reviews from Amazon and manually labeling the data by calculating the cosine similarly between the reviews and having human judges reviewing them. They applied the semantic language model (SLM) and SVM algorithm to compare the effectiveness of the models. According to the results, SLM outperformed the other methods by achieving an area under curve (AUC) score of 0.998, while SVM scored 0.557.

2.3 | Reviewer behavior-based detection models

Lim et al.[23] proposed a model based on the assumption that a spammer will target certain products or a brand to make the most impact. The rating he or she will give to the product will also be different from the average score provided by other reviewers. According to the authors, spammers will target and monitor few products, and then, they will write fake reviews when the time is appropriate to tip the value of the product rating; they rely on targeting products in a short period. The rating will be entirely different from other reviewers' ratings, and they will attempt to review products early to sway other reviewers' opinions. The evaluation results showed that the model is effective and outperforms other baseline methods based on helpful votes (ie, votes given by other users to the review).

Mukherjee et al.[24] built a so-called author spamicity model (ASM) to detect suspicious reviewers based on their behavior footprint. The idea is that reviewers can be categorized into 2 groups, spammers and nonspammers. Furthermore, each of these groups will have a different behavior distribution. Unlike previous papers on behavioral analysis, this is the first paper to propose detecting fake reviews using an unsupervised Bayesian inference framework. Also, it introduced a way to analyze features using posterior density analysis. A set of experiments involving human expert evaluation has been conducted to evaluate the proposed model. The results showcase that the proposed model is efficient and outperforms other supervised methods.

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Fei et al.[25] focused on detecting spammers who write reviews in short time windows. They argued that a burst of reviews can be either due to the sudden popularity of the product or a spam attack. They built a network of reviewers that appear in different bursts and then represented it in a graph (Markov Random Field). Furthermore, using statistical methods, they classified reviewers as spammers or not. The authors relied on behavior features, such as the rating deviation and reviewer burstiness; all features were normalized to [0,1] for simplicity. They achieved 77.6% accuracy with the proposed approach.

In contrast to the Mukherjee et al.[24] ASM model, Xie et al.[26] decided to focus on singleton reviews. According to their study of available data, more than 90% of reviewers write only 1 review. Also, the size of a singleton reviews is huge compared to nonsingleton reviews. Thus, these reviews can tip the rating of a product. They observed that spam attack arrival patterns are hectic compared to the stable arrival pattern of regular reviews. They explored the relationship between the volume of the singleton reviews and the rating of the store. Xie et al.[26] explained that if there is a sharp increase in the volume of singleton reviews and sharp fixation (low or high) of the store rating, spammers are attempting to manipulate a store's rating/reputation. First, for each time window, they calculated the ratio of singleton reviews, average rating, and the average number of reviews. They designed a multiscale anomaly detection algorithm on multidimensional time series based on curve fitting. The evaluation, which used a dataset involving 53 stores, indicated that the algorithm was able to detect fake stores with an accuracy of 75.86%.

Feng et al.[37] investigated the connection between the distribution anomaly and detecting fake reviews. They hypothesized that if a deceptive business hires spammers to write fake reviews, the spammers will try to tip the review score. Thus, they will distribute according to the natural distributions. For evaluation, they used a subset of the "gold-standard" dataset from Ott et al.,[16] which contains 400 deceptive and truthful reviews as training and testing data. They achieved an accuracy of 72.5% on their test data. The authors were also the first to provide a quantitative study that characterizes the distribution of opinions using data from Amazon and TripAdvisor. The proposed method is effective in detecting suspicious activity within a window of time. However, it is ineffective in detecting whether individual reviews are false or truthful opinions.

Li et al.[27] decided not to focus entirely on heuristic rules like previous researchers. They used manually labeled reviews from Epinions.com. They only focused on reviews that received high numbers of helpful votes and comments. They assumed that reviews with low helpful votes are more suspicious than reviews with a significant number of votes and are more likely fake. Furthermore, they ignored reviews from anonymous users and duplicated reviews. The authors used supervised learning techniques such as SVM, NB, and LR to identify review spam. Two groups of features were extracted. The first group consists of review features, such as content features (unigrams, bigrams), content similarity features, and sentiment features. The second group consists of reviewer-related features, such as profile features and behavior features. The NB method achieved much better results compared to other methods that rely on behavior features. It scored 0.583 F-score on all features. They observed that the score drops significantly when behavior features are excluded from the features set. They also noted that helpful votes perform poorly compared to other features.

2.4 | Fake news detection

Research on fake news detection is still in the early stage as this is a relatively recent phenomenon, at least in the interest raised by society. We review some of the published work in the following.

Rubin et al.[28] discuss 3 types of fake news. Each is a representation of inaccurate or deceptive reporting. Furthermore, the authors weight the different kinds of fake news and the pros and cons of using different text analytics and predictive modeling methods in detecting them. In their paper, they separated the fake news types into 3 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* is 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.

Rubin et al.[29] also proposed a model to identify satire and humor news articles. They examined and inspected 360 satirical news articles in mainly 4 domains, namely, civics, science, business, and what they called "soft news" ("entertainment/gossip articles"). They proposed an SVM classification model mainly using 5 features developed based on their analysis of the satirical news. The 5 features are *Absurdity*, *Humor*, *Grammar*, *Negative Affect*, and *Punctuation*. Their

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

highest precision of 90% was achieved using only 3 combinations of features, which are *Absurdity*, *Grammar*, and *Punctuation*.

Horne and Adali[17] 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 3 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[30] 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[31] introduced LIAR, a new dataset that can be used for automatic fake news detection. Although LIAR is considerably bigger in size, unlike other datasets, this dataset does not contain full articles; instead, it contains 12 800 manually labeled short statements from politicalFact.com.

All the existing approaches have focused on detecting only 1 type of fake content, either fake reviews or fake news, with the majority of the approaches targeted at fake reviews. To our knowledge, the approach presented in this paper is the first to effectively detect both fake news and fake reviews. Running our model on existing datasets and a new dataset indicates its ability to outperform related proposals.

3 | FAKE CONTENT DETECTION MODEL

3.1 | Approach overview

Our general approach for fake content detection is outlined in Figure 1. It starts with preprocessing the dataset by removing unnecessary characters and words from the data. The n-gram features are extracted, and a matrix of features 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 specifically investigated 6 different machine learning algorithms, namely, stochastic gradient descent (SGD), SVM, linear support vector machines (LSVM), K-nearest neighbor (KNN), LR, and decision trees (DT).

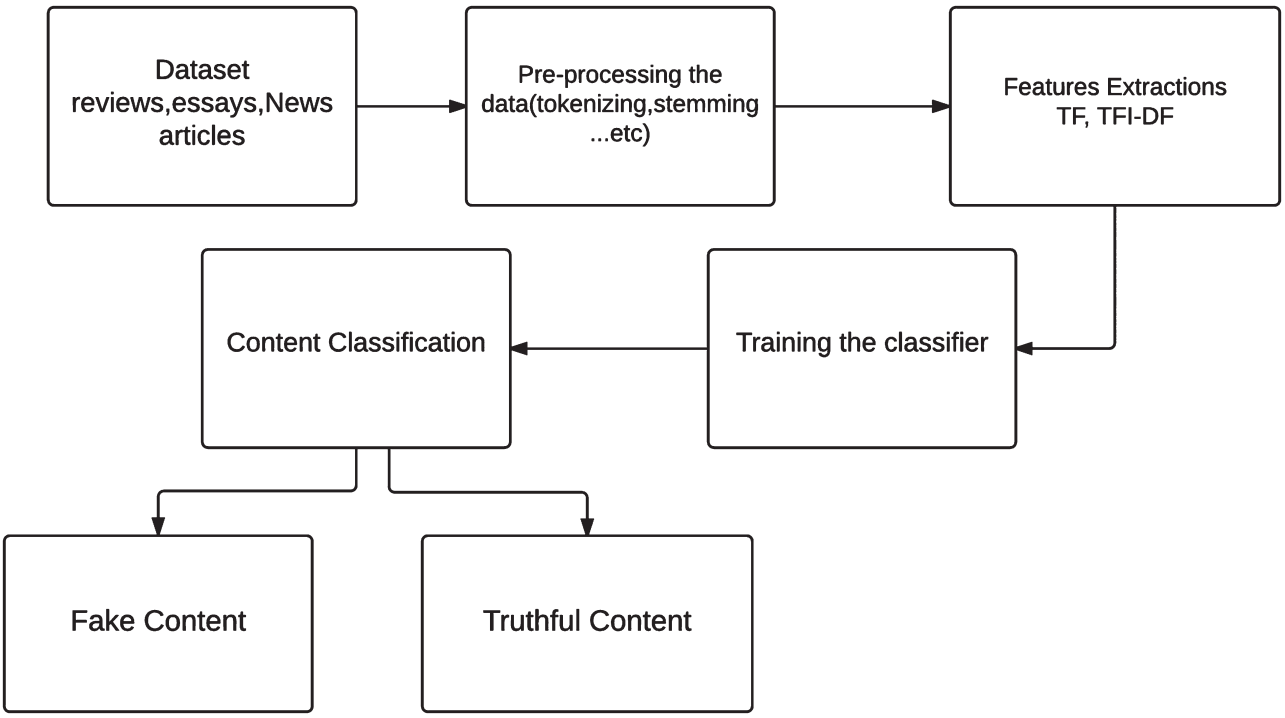

FIGURE 1 Classification process

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.2 | The n-gram model

The n-gram model is a popular feature identification and analysis approach used in language modeling and NLP fields.[32] The 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 study, we use a 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 content. The idea is to generate various sets of n-gram frequency profiles from the training data to represent fake and truthful content. We use several baseline n-gram features based on words and examine the effect of the n-gram length on the accuracy of different classification algorithms.

3.3 | Data preprocessing

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

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

3.4 | 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 frequently used 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*, and so on. Those words were removed from each document, and the processed documents were stored and passed on to the next step.

3.5 | Stemming

After tokenizing the data, the next step is to transform the tokens into a standard form. Stemming, put simply, is changing the words into their original form and decreasing the number of word types or classes in the data.[38] For example, the words "Running," "Ran," and "Runner" will be reduced to the word "run." We use stemming to make classification faster and efficient. We use the Porter stemmer, which is the most commonly used stemming algorithm due to its accuracy.

3.6 | 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 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. This research used 2 different features extraction methods, namely, *term frequency (TF)* and *term frequency-inverted document frequency (TF-IDF)*. These methods are described in the following.

3.7 | Term frequency

TF 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 word 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 a word existing in the documents. For example, if a word is in a certain document, it will be represented as 1, and if it is not in the document, it will be set to 0. Thus, each document is represented by groups of words.

Let D denote a corpus, or set of documents. Let d denote a document, $d \in D$; we define a document as a set of words w. Let $n_w(d)$ denote the number of times the word w appears in document d. Hence, the size of document d is $|d| = \sum_{w \in d} n_w(d)$.

The normalized TF for word w with respect to document d is defined as follows:

$$TF(w)_d = \frac{n_w(d)}{|\, d \,|}.$$

3.8 | Term frequency-inverted document frequency

TF-IDF is a weighting metric often used in information retrieval and NLP. It is a statistical metric used to measure how important a term is to a document in a dataset. A term's 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.

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The inverse document frequency (IDF) for a term w with respect to document corpus D, denoted $IDF(w)_D$, is the logarithm of the total number of documents in the corpus divided by the number of documents where this particular term appears, and is computed as follows:

$$IDF(w)_D = 1 + \log\left(\frac{|D|}{|\{d : D|w \in d\}|}\right).$$

One of the main characteristics of IDF is that it weights down the TF 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 lessens the impact of these terms.

TF-IDF for the word w with respect to document d and corpus D is calculated as follows:

$$TF - IDF(w)_{d,D} = TF(w)_d \times IDF(w)_D.$$

So, for example, we have a document with 200 words, and we need the TF-IDF for the word "people." Assume that the world "people" occurs in the document 5 times; then, TF = 5/200 = 0.025. Now, we need to calculate the IDF; let us assume that we have 500 documents, and "people" appears in 100 of them. Then, IDF(people) = 1 + log (500/100) = 1.69, and TF-IDF(people) = 0.025 × 1.69 = 0.04056.

3.9 | Classification

After extracting the features using either TF or TF-IDF, we train a machine learning classifier to decide whether a sample's content is truthful or fake.

Given a document corpus or dataset, 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.

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

Using a feature extraction technique (ie, 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 a value is null (0) if the term is not in the document.

Using the notation and definition given earlier:

- For TF: $feature(t_j) = TF(t_j)_{d_i}$
- For TF-IDF: $feature(t_j) = TF - IDF(t_j)_{d_i,\Delta}$

3.10 | Implementation

In our model, the TF will represent each term in our vector with a measurement that illustrates how many times the term/features occurred in the document. We use *Count Vectorizer* class from *scikit-learn*, a Python module, to produce a table of each word mentioned and its occurrence for each class. *Count Vectorizer* learns the vocabulary from the documents and then extracts the word's count features. Next, we create a matrix with the token counts to represent our documents.

We use *TfidfVectorizer* and *TfidfTransformer* from *scikit-learn* for implementation. *TfidfVectorizer* converts the documents into a matrix of TF-IDF features, while *TfidfTransformer* is used to normalize the TF-IDF representation.

We studied 6 different classifiers to predict the class of the documents, including SGD, SVM, LSVM, LR, KNN, and DT. We used implementations of these classifiers from the Python *natural language toolkit (NLTK)*.

4 | EXPERIMENTAL EVALUATIONS

In this section, we describe the experimental evaluation of our proposed approach and discuss obtained results.

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.1 | Experiments overview

We conducted 2 series of experiments: 1 to assess the ability of the proposed model to detect fake reviews and the other to assess its ability to detect fake news. Obviously, we used 2 different datasets in both experiments. However, the same procedure was followed in both cases.

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 ($n = 2$), and then steadily increased n by 1 until we reached $n = 4$. Furthermore, each n value was tested combined with a different number of features. In both experiments, we tested the n-gram features using 2 different feature extraction methods, namely TF and TF-IDF. All experiments were run using 5-fold cross validation; in each validation round, the dataset was divided into 80% for training and 20% for testing.

As mentioned earlier, we studied 6 different machine learning algorithms. 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.

4.2 | Experiment 1: Fake review detection

4.2.1 | Dataset 1

In this experiment, we used a public dataset, collected by Ott et al.,[16] which contains 800 truthful reviews and 800 fake reviews, all written in English. The truthful reviews are from TripAdvisor and cover the webpages of the 20 most popular hotels in Chicago. The fake reviews were written by 400 human-intelligence tasks (HITs) hired on Amazon Mechanical Turk. The HITs were asked to pretend that their boss asked them to write either positive or negative fake reviews for the hotels; 800 fake reviews were produced covering the same 20 hotels as the aforementioned truthful reviews from TripAdvisor. When collecting the reviews, Ott et al.[16] ignored reviews with less than 5 stars and reviews with less than 150 characters. For every review, the following information is available:

- Hotel name
- Review text
- Review sentiment (Positive or negative)
- Review label (Fake or truthful)
- Also, we added the review text length—the total number of words in the text.

4.3 | Data analysis

From our analysis of the above dataset, we found out that fake content contains more function words and content words than honest reviews, as illustrated in Figure 2.

- Function words are words with little lexical meaning, which are usually used to express a grammatical relationship with other words (e.g., *the, that, she, of*).
- Content words are the opposite, in the sense that they stand on their own by carrying specific purpose or meaning (not necessarily in connection with other words), for example, flew and mountain.

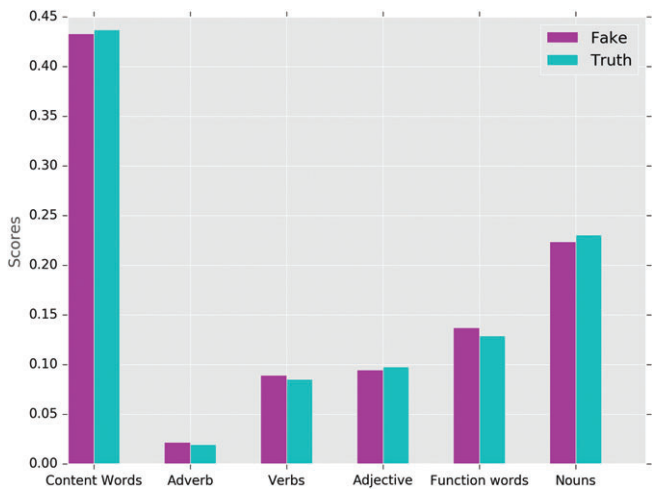

FIGURE 2 Word distribution in Dataset 1. Even though the difference is slight, it can be seen that fake content writers use more content and filler words. Furthermore, they use more verbs and adjectives than real content writers

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

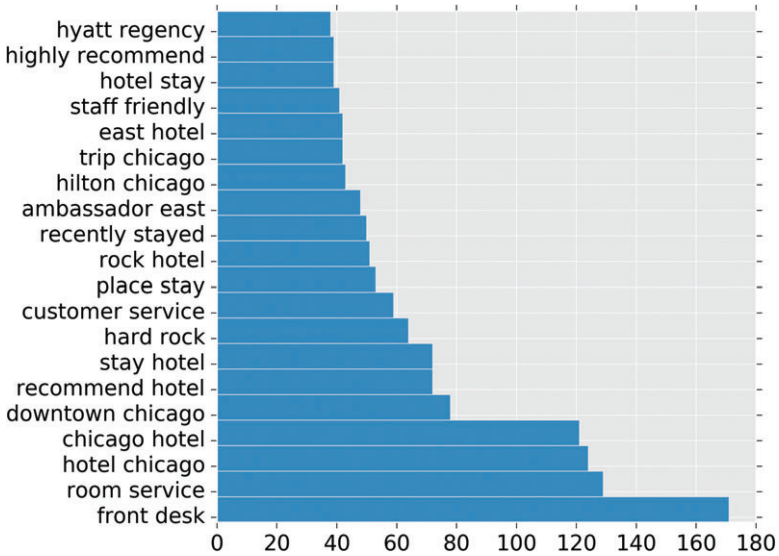

FIGURE 3 Top 20 bigram in fake reviews from Dataset 1

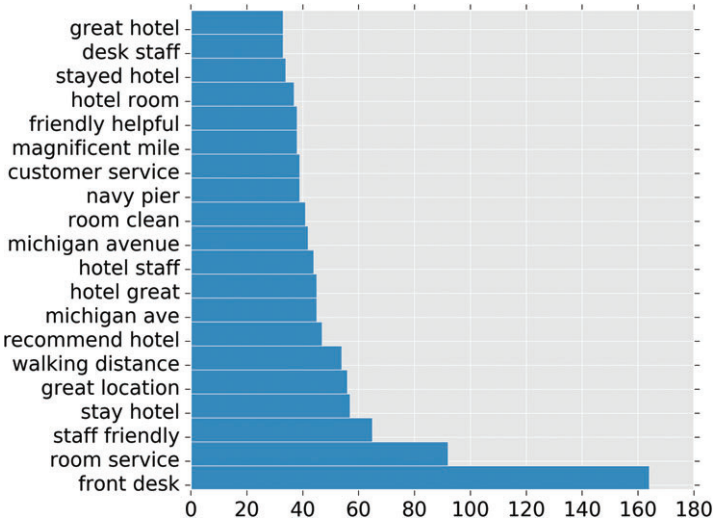

FIGURE 4 Top 20 bigram in real reviews from Dataset 1

This means that fake review writers tend to use more filler/functions and content words than real reviews writers. Furthermore, they use more verbs and adverbs than real reviews writers. In contrast, real reviews contain more nouns and adjectives.

Figures 3 and 4 depict the 20 most frequently used bigram in fake reviews and honest reviews from the dataset, respectively.

As can be seen, both depictions contain similar terms such as "front desk" and "room service." However, there are some key differences. For instance, fake reviewers tend to use the hotel's name more often, while honest reviewers tend to reference locations around the hotel's location to visit.

4.4 | Experiments results

We run the aforementioned algorithms on Dataset 1 with the goal of predicting whether the reviews are truthful or fake. We studied 2 different features extraction methods, TF-IDF and TF, and varied the size of the n-gram from $n = 1$ to $n = 4$.

We also varied the number of features p (ie, top features selected; see Section 3.5 for details), ranging from 1000 to 50 000. The prediction results for the different classification algorithms are depicted in Tables 1–6.

As can be seen from the above results, linear-based classifiers (linear SVM, SDG, and LR) achieved better results than non-linear ones. We obtained 90% accuracy using linear SVM. It is also noticeable that they achieve better accuracy with more feature values as they use 10 000 and 50 000 feature values. We can also observe that increasing the n-gram size lowers the accuracy when using all the classifiers. TF-IDF performed better than TF, which is expected given that fake and honest reviews contained similar words. The lowest accuracy of 47.2% was achieved using KNN with 4-gram words and 50 000 feature values.

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 SVM prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	83.0	82.0	83.0	82.0	83.0	82.0	79.0	82.0
Bigram	80.0	83.0	82.0	80.0	80.0	83.0	82.0	80.0
Trigram	73.0	78.0	79.0	75.0	75.0	78.0	77.0	75.0
Fourgram	76.0	69.0	68.0	47.0	70.0	69.0	68.0	42.0

TABLE 2 LSVM prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	85.0	85.0	87.0	86.0	82.0	84.0	84.0	84.0
Bigram	77.0	85.0	90.0	85.0	75.0	79.0	78.0	79.0
Trigram	74.0	80.0	79.0	80.0	76.0	73.0	74.0	75.0
Fourgram	71.0	75.0	77.0	75.0	71.0	71.0	70.0	70.0

TABLE 3 KNN prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	75.0	72.0	73.0	71.0	64.0	63.0	60.0	63.0
Bigram	75.0	77.0	78.0	77.0	55.0	57.0	56.0	57.0
Trigram	53.0	72.0	73.0	72.0	55.0	54.0	57.0	54.0
Fourgram	48.0	55.0	69.0	69.0	56.0	55.0	52.0	47.0

TABLE 4 DT prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	69.0	71.0	62.0	69.0	65.0	68.0	66.0	71.0
Bigram	71.0	70.0	69.0	71.0	71.0	73.0	73.0	73.0
Trigram	70.0	70.0	72.0	57.0	71.0	70.0	71.0	0.62
Fourgram	67.0	67.0	67.0	68.0	64.0	72.0	68.0	67.0

TABLE 5 SGD prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	85.0	85.0	85.0	86.0	85.0	83.0	82.0	82.0
Bigram	79.0	84.0	83.0	86.0	76.0	82.0	80.0	80.0
Trigram	71.0	78.0	80.0	82.0	69.0	78.0	80.0	81.0
Fourgram	72.0	76.0	78.0	76.0	71.0	75.5	76.0	77.0

TABLE 6 LR prediction accuracy (in %) for Dataset 1 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	85.0	85.0	87.0	87.0	84.0	84.0	82.0	83.0
Bigram	79.0	86.0	86.0	85.0	79.0	83.0	84.0	84.0
Trigram	76.0	78.0	80.0	79.0	74.0	79.0	79.0	79.0
Fourgram	72.0	77.0	77.0	76.0	72.0	79.0	77.0	76.0

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.5 | Experiment 2: Fake news detection

4.5.1 | Dataset 2

In contrast to the previous datasets, this dataset was entirely collected from real-world sources. We collected news articles from Reuters.com (News website) for truthful opinions. As for the fake news, they were collected from a fake news dataset on kaggle.com. The dataset consisted of fake news from unreliable websites 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 articles because these are currently the main target of (fake news) spammers. The news articles from both the fake and truthful categories occurred in the same timeline, specifically in 2016. Each of the articles is longer than 200 characters.

For every review, the following information is available:

- Article text
- Article type
- Article label (fake or truthful)
- Article title
- Article date

4.6 | Data analysis

Similar to experiment 1, the analysis of sample data corroborates the fact that fake content contains more function words and content words than honest articles, as depicted in Figure 5. Also akin to the first dataset, honest news articles contain more nouns and adjectives. The fake news contains more adverbs and verbs. However, in contrast to Dataset 1, fake writers in Dataset 2 used less verb and adjectives in their content than real content writers.

Figures 6 and 7 show the 20 most frequently used bigrams in fake and honest news from the dataset, respectively.

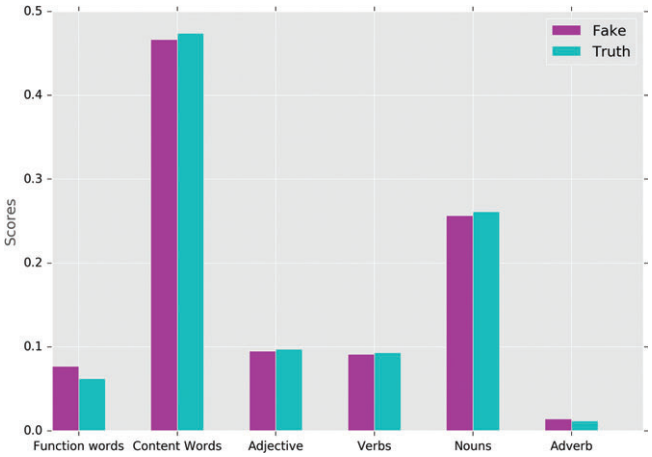

FIGURE 5 Word distribution in Dataset 2. Even though the difference is slight, it can be seen that fake content writers use more content and filler words. However, in contrast to Dataset 1, fake writers in Dataset 2 used less verbs and adjectives in their content than real content writers

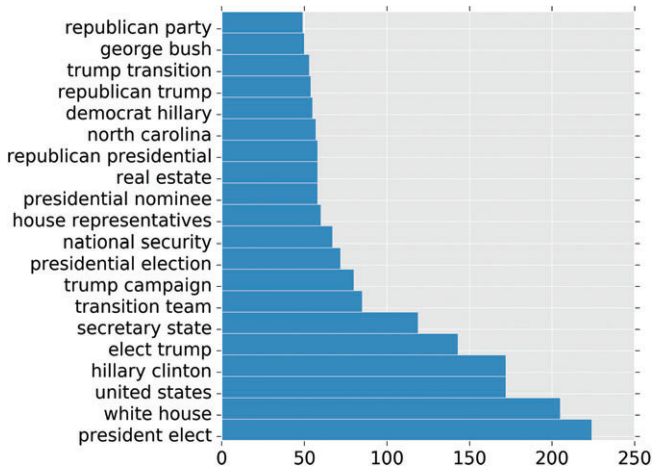

FIGURE 6 Top 20 bigram in real news from Dataset 2

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

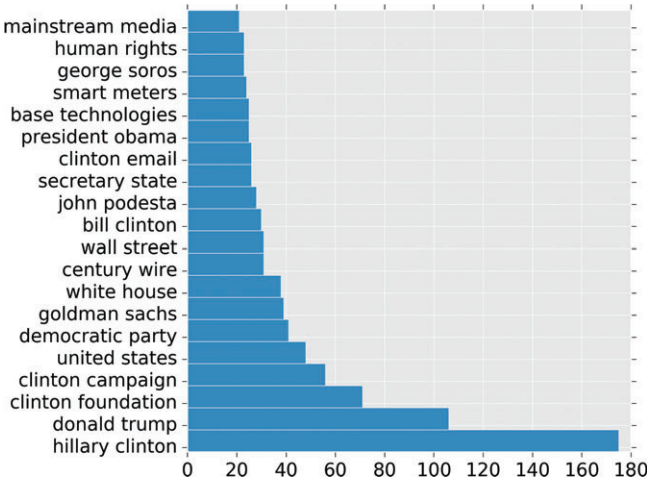

FIGURE 7 Top 20 bigrams in fake news from Dataset 2

As can be seen, both types of articles contain similar terms such as "Hillary Clinton" and "United States." However, in contrast to the first dataset, we see more variety of terms in both classes, even though we have targeted articles that cover similar incidents and topics.

4.7 | Experiments results

At 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 dataset so we can acquire 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 the real and fake articles we collected, 1000 fake articles and 1000 real articles. The 2000 articles represent a subset of the dataset described in the previous section that focuses only politics. We used the same approach as the previous experiment by running the algorithms on Dataset 2, with the goal of predicting whether the articles are truthful or fake. Tables 7–12 show the obtained results.

Similar to the results obtained in Experiment 1, linear-based classifiers (Linear SVM, SDG, and LR) 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, 92%. This classifier performs well no matter the number of feature values used. With the increase of n-gram (trigram, 4-gram), the accuracy of the algorithm also decreases.

However, trigram features still performed considerably better on this dataset. It also contains wider varieties of words. Furthermore, TF-IDF outperformed TF. The lowest accuracy achieved is 45% when using KNN with a features size of 50.000.

TABLE 7 SVM accuracy (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	84.0	86.0	84.0	84.0	85.0	72.0	69.0	69.4
Bigram	78.0	73.0	67.0	54.0	68.0	51.0	47.0	47.0
Trigram	71.0	59.0	53.0	48.0	53.0	47.0	53.0	47.0
Fourgram	55.0	37.0	37.0	45.0	47.0	48.0	40.0	47.0

TABLE 8 LSVM accuracy results (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	89.0	89.0	89.0	92.0	87.0	87.0	87.0	87.0
Bigram	87.0	87.0	88.0	89.0	86.0	83.0	82.0	82.0
Trigram	84.0	85.0	86.0	87.0	86.0	84.0	84.0	79.0
Fourgram	71.0	76.0	76.0	81.0	70.0	70.0	70.0	61.0

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 9 KNN accuracy (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	79.0	83.0	82.0	83.0	77.0	70.0	68.0	68.0
Bigram	67.0	65.0	68.0	64.0	62.0	55.0	51.0	45.0
Trigram	73.0	68.0	65.0	67.0	76.0	63.0	57.0	46.0
Fourgram	69.0	68.0	68.0	58.0	67.0	54.0	56.0	43.0

TABLE 10 DT accuracy (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gramsize	**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
Fourgram	74.0	74.0	71.0	74.0	67.0	67.0	70.0	67.0

TABLE 11 SGD accuracy (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gramsize	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	88.0	86.0	88.0	89.0	87.0	86.0	89.0	85.0
Bigram	86.0	85.0	87.0	86.0	85.0	84.0	85.0	84.0
Trigram	84.0	85.0	86.0	86.0	85.0	85.0	87.0	87.0
Fourgram	70.0	72.0	74.0	80.0	72.0	73.0	72.0	78.0

TABLE 12 LR accuracy (in %) for Dataset 2 for different features size (second row) and n-gram size

	TF-IDF				TF			
n-gram size	**1000**	**5000**	**10 000**	**50 000**	**1000**	**5000**	**10 000**	**50 000**
Unigram	83.0	89.0	89.0	89.0	89.0	89.0	83.0	89.0
Bigram	87.0	87.0	88.0	88.0	87.0	85.0	86.0	86.0
Trigram	86.0	85.0	88.0	87.0	83.0	83.0	83.0	82.0
Fourgram	70.0	76.0	75.0	81.0	68.0	67.0	67.0	61.0

4.8 | Performance comparison

As mentioned earlier, to our knowledge, our work is the first to assess the ability to detect both fake news and fake reviews, which—as shown earlier (from the analysis of the datasets)—have commonalities but also important differences. In both cases, our proposed model achieves very encouraging performance. Furthermore, our model performs well on the specific categories of fake content when compared with previous models. Comparison of previous works and our results for opinion spam and fake news detection models are presented in Table 13. As noted, our approach outperforms most of the existing work.

By running our model on Ott et al's review dataset, we achieve slightly better results (90%) than their best results of 89%. Furthermore, we conducted additional experiments by running our model on the news dataset of Adali and Horne,[17] consisting of real news from BuzzFeed and other news websites and satires from Burfoot and Baldwin's satire dataset.[30] We obtained 87% accuracy using n-gram features and the LSVM algorithm when classifying fake news against real news, which is much better than the 71% accuracy achieved by the authors on the same dataset.

5 | CONCLUSIONS

In recent years, deceptive content such as fake news and fake reviews have increasingly become a dangerous prospect for online users. Fake reviews have affected consumer and stores alike. The use of hired writers to produce fake reviews to boost sales

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 13 Comparison of previous works and our work for opinion spam and fake news detection

Dataset	Classifier	Features	Performance metrics	Score	Reference
Reviews Amazon website	Logistic regression	Review and reviewer features	AUC	78%	8
Dataset 1 (Ott et al. reviews dataset)	SVM	LIWC + Bigrams	Accuracy	89%	16
Dataset 1 (Ott et al. reviews dataset)	SVM	Stylometric features	F-measure	84%	16
Dataset 1 (Ott et al. reviews dataset)	LSVM	Bigram	Accuracy	90%	Our Results
Dataset 2 (Our news dataset)	LSVM	Unigram	Accuracy	92%	Our Results
Buzzfeed news and random new articles (Horne and Adali's news dataset)	SVM	Text-based features	Accuracy	71%	17
Buzzfeed news and random new articles (Horne and Adali's news dataset)	LSVM	Unigram	Accuracy	87%	Our results

is also growing. It becomes difficult for consumers to recognize good products from bad ones based on reviews. Furthermore, the problem of fake news has gained attention in 2016, especially in the aftermath of the last U.S. presidential election. Recent statistics and research show that 62% of U.S. adults get their news from social media.[33,34] Most of the popular fake news stories were more widely shared on Facebook than the most popular mainstream news stories.[35] A sizable number of people who read fake news stories have reported that they believe them more than news from mainstream media.[35] Dewey[36] claims that fake news played a huge role in the 2016 U.S. election and that it continues to affect people's opinions and decisions.

Opinion spam and fake news are a closely related phenomenon as both consist of writing and spreading false information or beliefs. The opinion spam problem was formulated for the first time a few years ago, but it has quickly become a growing research area due to the abundance of user-generated content. It is now easy for anyone to either write fake reviews or write fake news on the web.

In this research, we focused on the problem of detecting opinion spam and fake news using n-gram analysis through the lenses of different features extraction methods. The n-gram features performed well on real-world data and pseudo data. Furthermore, they performed better when applied on the fake news data. The type of data used for training and the size of features has an effect on the classifier performance; as observed, with the increase of the features, the majority of the classifiers achieved a higher accuracy. The majority of our higher accuracy was achieved using 50 000 and 10 000 features. We also saw that an increase in the n-gram size would cause a decrease in the accuracy. In both datasets 1 and 2, the unigram and bigram performed better than the trigram and quadgram.

In our future work, we would like to incorporate in our model, statistical features, and features that reflect the writer's styles such as the number of slang words or filler word in the text. Reviews and articles are a primary source of information. We can generate a huge number of features based on the text, which consequently will lead to high dimensionality. Thus, to tackle this problem, we would like to explore the effect of feature selection techniques such as information gains and chi-square.

It is hard to procure accurately labeled real-world datasets. Thus, we would like to explore unsupervised and semi-supervised methods to detect deceptive content. In addition, we would like to explore incorporate semantic features, and text complexity features to the n-gram features and testing such models on the datasets.

ORCID

Issa Traore http://orcid.org/0000-0003-2987-8047

REFERENCES

1. Anderson M, Anderson M. 88% of consumers trust online reviews as much as personal recommendations. *Search Engine Land* 2017. http://searchengineland.com/88-consumers-trust-online-reviews-much-personal-recommendations-195803. Accessed June 15, 2016.
2. BBC News. Samsung probed in Taiwan over 'fake web reviews'. *BBC News* 2017. http://www.bbc.com/news/technology-22166606. Accessed June 15, 2016.
3. Helft M. Reverb P.R. firm settles case on fake reviews. *Nytimes.com*. 2017. http://www.nytimes.com/2010/08/27/technology/27ftc.html. Accessed June 15, 2016.
4. Li J, Ott M, Cardie C, Hovy E. 2014. Towards a general rule for identifying deceptive opinion spam. Paper presented at: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics; June 23-25, 2014:1566–1576; Baltimore, MD: ACL
5. Miller C. Cosmetic surgery company settles case of faked reviews. 2017. *Nytimes.com*.
6. Mukherjee A, Liu B, Glance N. Spotting fake reviewer groups in consumer reviews. Paper presented at: Proceedings of the 21st international conference on World Wide Web; 2012. Lyon, France: ACM.

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

7. 26NEWS. Woman paid to post five-star google feedback. 2017. http://www.thedenverchannel.com/news/woman-paid-to-post-five-star-google-feedback. Accessed June 15, 2016.
8. Jindal N, Liu B. *Opinion spam and analysis*. *Proceedings of the 2008 International Conference on Web Search and Data Mining*. New York, NY: ACM; 2008:219-230. https://doi.org/10.1145/1341531.1341560.
9. Gu L, Kropotov V, Yarochkin F. The fake news machine, how propagandists abuse the internet and manipulate the public. 2017. 1st ed. [pdf] Trend Micro, p. 81. https://documents.trendmicro.com/assets/white_papers/wp-fake-news-machine-how-propagandists-abuse-the-internet.pdf?_ga$=$2.117063430.1073547711.1497355570-1028938869.1495462143. Accessed January 1, 2017.
10. 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. Accessed January 1, 2017.
11. 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 January 1, 2017.
12. Schow A. The 4 types of 'Fake News'. *Observer*. 2017. http://observer.com/2017/01/fake-news-russia-hacking-clinton-loss/. Accessed January 26, 2017.
13. 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. Accessed January 1, 2017.
14. 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 January 1, 2017.
15. Ye S, Shyhtsun Felix W. Measuring message propagation and social influence on twitter. Com. *SocInfo*. 2010;10:216-231.
16. Ott M, Choi Y, Cardie C, & Hancock JT. (2011, June). Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 309–319). Association for Computational Linguistics. Accessed December 20, 2017.
17. Horne BD, 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. Paper presented at: The 2nd International Workshop on News and Public Opinion at ICWSM; Montreal, Canada 2017; Montreal, Canada.
18. Mukherjee A, Venkataraman V, Liu B, Glance N. Fake review detection: classification and analysis of real and pseudo reviews. UIC-CS-03-2013. Technical Report; 2013.
19. Mukherjee A, Venkataraman V, Liu B, Glance N. What yelp fake review filter might be doing. Paper presented at: Proceedings of the International Conference on Weblogs and Social Media; Boston, USA 2013, Boston, USA.
20. Shojaee S, Murad MAA, Azman AB, Sharef NM, Nadali S. Detecting deceptive reviews using lexical and syntactic features. Paper presented at: Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference; December, 2013:53-58; Seri Kembangan, Malaysia: IEEE.
21. Algur SP, Patil AP, Hiremath PS, Shivashankar S. Conceptual level similarity measure based review spam detection. Paper presented at: Signal and Image Processing (ICSIP), 2010 International Conference; December, 2010:416-423; Chennai, India: IEEE.
22. Lau RY, Liao SY, Kwok RCW, Xu K, Xia Y, Li Y. Text mining and probabilistic language modeling for online review spam detecting. *ACM Trans Manage Inf Syst*. 2011;2(4):1-30.
23. Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW. Detecting product review spammers using rating behaviors. Paper presented at: Proceedings of the 19th ACM international conference on Information and knowledge management; October, 2010:939-948; Toronto, Canada: ACM.
24. Mukherjee A, Kumar A, Liu B, Wang J, Hsu M, Castellanos M, Ghosh R. Spotting opinion spammers using behavioral footprints. Paper presented at: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining; August, 2013:632-640; Chicago, IL: ACM.
25. Fei G, Mukherjee A, Liu B, Hsu M, Castellanos M, Ghosh R. Exploiting burstiness in reviews for review spammer detection. Paper presented at: International Conference on Web and Social Media (ICWSM); 2013, Vol. 13:175-184. Boston, MA.
26. Xie S, Wang G, Lin S, Yu PS. Review spam detection via time series pattern discovery. Paper presented at: Proceedings of the 21st International Conference on World Wide Web; April, 2012:635-636; Lyon, France: ACM.
27. Li F, Huang M, Yang Y, Zhu X. Learning to identify review spam. Paper presented at: IJCAI Proceedings-International Joint Conference on Artificial Intelligence; July, 2011, Vol. 22, No. 3, p. 2488, Barcelona, Spain.
28. Rubin VL, Chen Y, Conroy NJ. Deception detection for news: three types of fakes. Paper presented at: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community (ASIST '15); 2015; Silver Springs, MD: American Society for Information Science, Article 83, 4 pages.
29. Rubin VL, Conroy NJ, Chen Y, Cornwell S. Fake news or truth? Using satirical cues to detect potentially misleading news. Paper presented at: Proceedings of NAACL-HLT; 2016:7-17, San Diego, CA.
30. Burfoot C, Baldwin T. Automatic satire detection: are you having a laugh? Paper presented at: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, August 04-04, 2009; Suntec, Singapore.
31. Wang WY. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. arXiv preprint arXiv:1705.00648. Vancouver, Canada.
32. Shannon CE. A mathematical theory of communication. *ACM SIGMOBILE Mobile Comput Commun Rev*. 2001;5(1):3-55.
33. Gottfried J, Shearer E. News use across social media platforms. *Pew Research Center, 26*, 2016.
34. Gottfried J, Barthel M, Shearer E, Mitchell A. The 2016 presidential campaign–a news event that's hard to miss. *Pew Research Center, 4*, 2016.
35. Silverman C, Singer-Vine J. Most Americans who see fake news believe it, new survey says. *BuzzFeed News* 2016.
36. Dewey C. Facebook has repeatedly trended fake news since firing its human editors. *Washington Post* 2016.
37. Feng S, Xing L, Gogar A, Choi Y. Distributional footprints of deceptive product reviews. Paper presented at: International Conference on Web and Social Media (ICWSM);2012, Vol. 12:98-105, Dublin, Ireland.
38. Mason J.E, Shepherd M, and Duffy J, "Classifying Web Pages by Genre: An n-Gram Approach", IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies. WI-IAT '09, 15–18 Sept. 2009, Milan, Italy, Italy.

How to cite this article: Ahmed H, Traore I, Saad S. Detecting opinion spams and fake news using text classification, *Security and Privacy*, 2018;1:e9. https://doi.org/10.1001/spy2.9

24756725, 2018, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spy2.9 by Massachusetts Institute of Technolo, Wiley Online Library on [30/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License