Fake News Detection using Machine Learning

Graham Healy, Nelson Blickman, Nijo Jacob

Abstract

In recent years, the booming development of social media and the progression of fake news have become a major concern due to its ability to create devastating impacts. Many research efforts have been made aiming to understand fake news and identify features and patterns of fake news. In the context of detecting fake news, machine learning methods has been used for this research purpose. In this paper, we use machine learning to conduct fake detection. We use supervise learning method to detect fake news by focusing on headline of news articles. This research analyzes the headlines of fake news, by the classification algorithm naïve bayes model. This outperforming better than existing methods up to >95% in unigram feature and unigram including sentence length combined features.

1. Introduction

Over the past decade, the social media systems have significantly changed the way users interact and communicate online. With the spawning of new applications that reshaping the existing news has influenced the way it is produced, dispersed and consumed within our society. The issue of disinformation on the internet has sky-rocketed, and it can potentially affect news readers opinions on critical issues and topics in the society. The use of misinformation, lies, and deceit has been around forever, but incorporating them and social media into fake news has turned it into a world phenomenon issue. From hackers and con artists to political opponents, fake news authors and publishers have engaged in the activity of creating and spreading false statements and conspiracy theories for their own personal interests or political gain. The issue is relevant, pressing, significant, and is nothing short of a serious concern.

Disinformation leads to a variety of issues. These issues include starting a genocide, war, and racism, which are few candidates that have at times been cultivated by propaganda that is false and misleading. Not only that misinformation has become a great threat to democracy, journalism and economies. It has weakened the public trust in government and undermines important values. Take the recent US elections as an example. If there is a conspiracy theory circling the web about a political opponent that isn't true, it impedes on a voter's right, in the sense that they are being misdirected but do not know it. Fake news, in fact, often spreads faster and deeper than real, credible news [7]. With all of this in mind, we have great motivation for contributing successful, careful research to this topic.

With fake news and disinformation being a problem for so long, many approaches have surfaced to combat it. Professional human fact checkers (Snopes, Washington Post, New York Times, etc.) have worked against disinformation campaigns for decades, and they have been effective. Unfortunately, this approach is time and energy consuming [6], and in the age of social media, it can fight only the tip of the iceberg. Further, while these fact checkers and other analysts offer a complete breakdown of fake news, they may be wasting their time, since many readers will not take the time to read their complete report. According to the Center for Information Technology and Society at the University of California Santa Barbara (UCSB), the headline is the most important point of emphasis for disinformation publishers and authors.

Additionally, a common technique in using supervised learning is to analyze the author's profile or website's history, like the approach used by Zhuk, Tretiakov, and Gordeichuk [9]. While this is an important dimension of the fake news and disinformation dilemma, it can also act as a double-edged sword. The Center for Information Technology and Society at UCSB explains how easy and cheap creating and maintaining a website is, and they detail how authors and publishers will imitate real news websites to appear credible. Since websites may pop up at any time, and since content can be posted anonymously, we decided to focus our research exclusively on the headlines of our news article data. We address our approach in a later section of this paper.

Having the ability to distinguish between a webpage that is either false or factual is an important step. From there, the page can be removed, labelled, or even blocked from arising in the first place. Therefore, our solution which will contribute to the ability to locate this disinformation on the internet will be profound for all of the reasons listed above.

2. Background

We begin by looking at important information on the structure of the classification algorithm naïve bayes and the types of features utilize in running fake news detection.

2.1. Naïve bayes

Naïve Bayes classifier is widely used in the field of machine learning, a powerful tool in measuring probability. It was originated comes from the Bayes Theorem. The Bayes Theorem describes the probability of an event supported by prior knowledge of the condition which is related to the event [2]. Here we show a mathematical representation of Bayes theorem: P(A|B) = P(B|A) *P(A) / P(B). The A and B are different events and P(B) is condition set not equal to zero. The conditional probability of P(A|B) is the likelihood of even A occurs if given B is true. It is measured by P(B|A) a conditional probability measure likelihood of even B occurs if given A is true times by P(A) divided by P(B). Both P(A) and P(B) are marginal probability (observing A and B).

Bases on the Bayes Theorem, the classifier model is created by put a naïve assumption to the Bayes theorem. Changing it from dependent conditional to an independent conditional probability model and simplifies the calculation for independence among the features. The model considers each input variable as being independent of each other, So, for any two events A and B are independent then P (A, B) = P(A)P(B). In addition to building the basic classifier, we could also address the problem of underflow and smoothing when implementing the model for fake news detection.

2.2. Features of fake news

To confront the fake news problem, we tend to focus on how to comprehend better of this issue with techniques and features. With modern technology, fake news tends to spread faster than real news. We focus on examining how to aim better at detecting fake news. We looked at existing attempts used to test the problem, the various solutions to the problem, and the discussion of typical patterns that can be used for features for the problem. For example, in the "Automatic Detection of Fake News", they run a set of learning tests to build an accurate fake detector using linguistic feature [5]. This shows the previous works of using specific data and features to train classifiers tend to focus without explicit reasoning and focus on typical patterns seen when detecting fake news.

We also see other features that are used in identifying fake news, many of these techniques were focused on extracting news contents and using the language features such as words and phrases from the source. For instance, Shu et al, illustrates these various language structures (Syntax) such as such as bag of words as n-gram and part of phrases of the sentences [6]. Another way of identifying fake news was extracting the source of the news, capturing the credibility of the news based on the popularity or trustworthiness of domains such as Facebook, BuzzFeed that publish these news articles. Based on these existing approaches, we intended to deviate away from the content and focus on extract news headlines than content or source and use sentence features (n-grams) to classify fake from realistic news.

3. Approach

Clearly, accurately classifying when an article is fake news or true information is beneficial, and our Naïve Bayes classifier that we created does just that. To create this classifier, we began by uploading available datasets from Kaggle. The data includes the text of a particular article, the title (or heading) of that article, and whether or not this article was fake or true. Hence, we will be using supervised Machine Learning.

After loading this data into our Python Jupyter Notebook, we take the first 75% of the articles for training. We are going to train our algorithm on the article title, and not the body of text within the article. Therefore, we next loop through every word of every title. If the word appeared in the title of a fake news article it would be appended to our fake news dictionary, and the same goes for words of "true" articles. (We split our title into individual unigram words by using tokenization.) From here, we loop through our two created dictionaries in order to create Counter structures that will count how many times a particular word appears in that dictionary, and thus appears in all the titles of fake (or true) articles, which allows us to calculate our Naïve Bayes probabilities more easily in the next section.

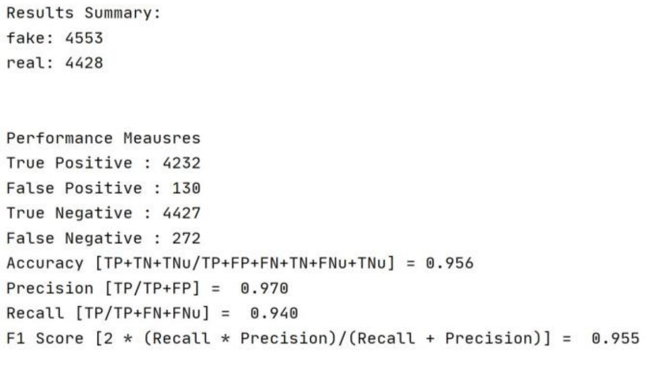
It is time to take the other 25% of the article titles to test a classifier. We loop through these article titles and for each title we tokenize every word within it. Then we use Naïve Bayes to find the likelihood that the article title belongs to a fake article and its likelihood it belongs to a true article. Our Naïve Bayes classifier will assume the article title is associated with the class with the higher probability (likelihood). In order to calculate our probability for a particular article title belonging to a certain class, we find the probability that a word in the title belongs to a certain class (fake or true), do this for each word in the title, and then take the product of those probabilities. If the probability of a particular word being in a class comes out to 0, we set that probability equal to Machine Epsilon for reasons explained in our results below. This will give us what Naïve Bayes calls, the Posterior probability. Then, we take the simple Prior Probability (articles of a class over total number of articles) and multiply it with the Posterior. Finally, we see whether our Naïve Bayes probability matches the true classification of that article.

In addition to our current classifier, we also build an another Naïve Baye classifier using the same dataset from Kaggle. After loading the data, we took 80% of articles for training and 20% for test. Trian the words just like the previous classifier loop through every words of every title, tokenize text into a array of words and append to key value dictionary to either positive or negative words and count the number of articles in positive and negative articles. Also taking the summation of all positive and negative word counts. Unlike our original we created this classifier uses a different method. This Naïve Bayes classifier first calculate the probability of all positive and negative text. By first taking the log of one times all positive text over the sum of all positive and negative text and you do the same thing with negative text. The classifier then adds on from previous probability of positive or negative text to each individual probability positive or negative words. The individual probability is calculated by taking log of one plus positive or negative word over total count positive or negative words. The summation of the logs of the probabilities is used to prevent underflow and adding one smoothing to each word to prevent from throw off the calculation.

We also test tried our second classifier with additional feature sentence length combined with the unigram. Train each individual words and length of the text and include in positive and negative dictionary. Finally, taking the test set and calculating the probability of each positive and negative length using same method of the classifier. The classifier calculated the probability of total text, each word in the text and length of text for each real and fake articles. Based on both classifiers, we stick to the original as it produces higher accuracy compare to second classifier.

4. Evaluation

Our results are phenomenal. The classifier's precision, accuracy, Recall, and F1_Score all come out to greater than 99%. We continued to test our classifier on articles that were not in the dataset we uploaded as well, and the results were still very impressive. Our classifier was trained on 40,000 labelled articles, so it had plenty of information to learn from. In fact, we anticipated a lower success rate. One would think that there are too many words that can show up in either the True information or Fake News. Our results show us though that there are enough distinct words that allow us to differentiate and classify the text. This is a success, and I am sure any company would be very happy with this success rate.



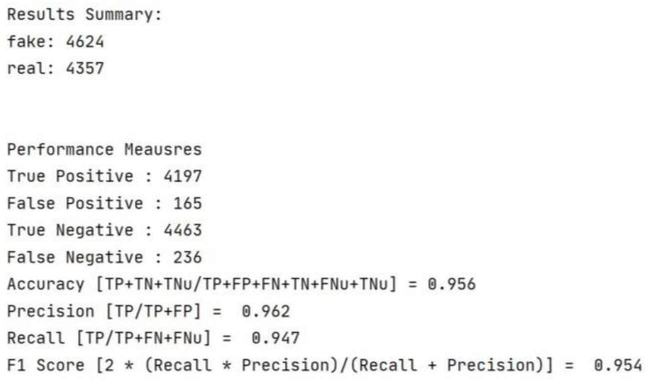
```
Results Summary:
fake: 4553
real: 4428


Performance Meausres
True Positive : 4232
False Positive : 130
True Negative : 4427
False Negative : 272
Accuracy [TP+TN+TNu/TP+FP+FN+TN+FNu+TNu] = 0.956
Precision [TP/TP+FP] =  0.970
Recall [TP/TP+FN+FNu] =  0.940
F1 Score [2 * (Recall * Precision)/(Recall + Precision)] =  0.955
```

Figure 1. Analysis of Unigrams

While the result from our second classifier containing the same number of labelled articles was almost close to our original classifier. The classifier performance measure each of the following category were precision 95.6%, accuracy: 96.2%, Recall: 94.7% and F1_Score 95.4% (see Figure 1). There were few amounts of false positive about 100 text compare to true positive about 4000 text and false negative about 200 text compare to true negative about 4000 text. This is due to calculation of the classification in term of summation of probability of each words and text class (positive, negative).

In addition, within the same classifier we tested an addition features of sentence length combined with unigram was almost exactly as same as the unigram. Only a slight difference such as the increase within precision from 96 % to 97% and F1 score and Recall performance measure (see Figure 2). Though there are difference from the original classifier, the result was still successful as our original classifier greater than >94%.



```
Results Summary:
fake: 4624
real: 4357


Performance Meausres
True Positive : 4197
False Positive : 165
True Negative : 4463
False Negative : 236
Accuracy [TP+TN+TNu/TP+FP+FN+TN+FNu+TNu] = 0.956
Precision [TP/TP+FP] =  0.962
Recall [TP/TP+FN+FNu] =  0.947
F1 Score [2 * (Recall * Precision)/(Recall + Precision)] =  0.954
```

Figure 2. Analysis of Unigrams + Sentence

This original classifier does not do anything innovative. However, a Naïve Bayes classifier can be built in a number of different ways. For example, companies build their classifiers in ways to prevent overfitting and smoothing issues. This may include summing up the logs of the probabilities as to opposed to taking the product in the posterior calculation. Our classifier does not do this. Instead of using 0 as a probability if a word does not appear in its class, our classifier uses Machine Epsilon, which is just enough to make sure that the posterior probability does not just come out to be 0.

Of course, as we will discuss below when we talk about our other related work, our classifier will not solve every existing problem. The test that our classifier went through involved articles that may have had a stronger indication that they were fake or true. However, people who create disinformation understand this, and adapt to create new disinformation article titles that are similar to the titles of true articles. Our classifier does remarkably well and shows that other classifiers that have lower accuracy can do better, but this does not solve the entire issue of disinformation.

5. Other Related Work

In 2015, a group of three used Natural Language Processing in order to detect fake news. Techniques included classifying the text within an article by using network analysis, sentiments, behavioral information, and linguistic features. They used 50,000 features to achieve an accuracy of about 90%. Many people adopted this approach, adding features such as "bigrams" (we used unigrams) and more. Another approach by researchers in 2017 used Random Forest Classifiers and SVM, which achieved an f-score of 65%. The attempts continued, and in 2017, along with basic NLP processes, deep learning and convolutional neural networks were being used.

The approach to classifying fake news has moved to include features that go beyond the text of the article itself, which include the authenticity of the source and the style of that article. In 2019, a classifier was built using two modules. The first one uses a knowledge base to retrieve articles from in order to verify the truth in an article, and the second one uses deep neural networks to train and predict what will be fake content. The accuracy of the combination of these models goes above 80 percent.

This accuracy rate is lower than ours. However, these models discussed above are much more complex than ours, using more features and machine learning techniques. I'm sure that this algorithm must correctly capture things that ours would miss. Yet it is worth pointing out that the idea of classifying the "title" of the article in order to understand the class of the article, which is what we did, may be undervalued in the current approaches. Our classifier may show at the very least, that there is a feature that these incredibly complex algorithms are forgetting to take into account, at least to a degree.

Other research approaches have utilized authors' and websites' profiles in their analysis of whether content is fake news. Specifically, Zhuk, Tretiakov, and Gordeichuk [9] implemented a model in their paper, "Methods to Identify Fake News in Social Media Using Machine Learning." Their model had a 70% accuracy in evaluating text to be factual or not, and it analyzed writing style and classification of sources and authors. This is valuable information, and it should be considered in forming a model. We considered using it for this research paper, but we ultimately decided to focus exclusively on titles because of author anonymity on the internet.

Further, our decision to focus solely on headlines was reinforced by Vosoughi, Roy, and Aral's findings in "The spread of true and false news online" about how fake news really spreads on social media. In their conclusion, the authors reiterate that although much scrutiny has been on social media bots, humans are the primary spreaders of false information on the internet [7]. Vosoughi, Roy, and Aral go further to say that many of the attempts to use machine learning to identify fake news have been implemented ad hoc. Therefore, we wanted to carry out our research in a with the headline, a specific element of every important news and fake news article.

6. Conclusion

This research paper confirmed some hypotheses of the project that our group members had but revealed some surprising results. It reinforced our belief of the importance of headlines, and we believe that they should not be overlooked in any fake news analysis. In addition, it reiterated the importance of not only thorough analysis, but also highlighted the importance of asking the right questions to begin with. While "one size fits all" is not applicable for all approaches, continuing ad hoc is also not optimal, as Vosoughi, Roy, and Aral have detailed.

At the very least, the contributions of this research enable further analysis of titles, subtitles, and other keywords using replication of Naïve Bayes or a different method. With our model's accuracy being as high as it is, it may be suitable to analyze whether our approach and approaches like ours ask the correct questions and evaluate the right variables. For example, if researchers believe that being able to identify the body of text as fake news is groundbreaking, it may also be beneficial to justify why this approach is important. Does identifying a body of text accomplish the goal if readers are not even looking through it?

Or, in the context of our paper, even if our accuracy is high, are we asking the right questions? Is our focus (headlines) in the right place, or should we be looking elsewhere? How can we, or anyone take the next step if we are not addressing the root of the problem? While we have identified previous research to justify taking our approach, nailing down the root causes of fake news spread (and how to spot it) definitively is a great next step, and we believe it much better than an ad hoc approach.

Finally, there will be challenges ahead in this field, even with successful analysis and answers to the questions posed in the last paragraph. Fake news and disinformation are adversarial topics, meaning that authors and publishers of that content will be looking to slip by models undetected. If our model gains traction, and scrutiny on headlines increases, authors and publishers of disinformation will alter the wording and punctuation of their headlines. In a way, this could be a good thing, because driving the misleading titles to be worded more like credible titles is a major milestone. However, the fake news dilemma is unlikely to be solved by adversaries changing their wording. Researchers and analysts should continue to look for better approaches, especially in the age of social media.

In 2015, a group of three used Natural Language Processing in order to detect fake news. Techniques included classifying the text within an article by using network analysis, sentiments, behavioral information, and linguistic features. They used 50,000 features to achieve an accuracy of about 90%. Many people adopted this approach, adding features such as "bigrams" (we used unigrams) and more. Another approach by researchers in 2017 used Random Forest Classifiers and SVM, which achieved an f-score of 65%. The attempts continued, and in 2017, along with basic NLP processes, deep learning and convolutional neural networks were being used.

7. Bibliography

[1] Burston, Adam, et al. "Where Does Fake News Come From?" Center for Information Technology and Society - UC Santa Barbara, 28 Aug. 2018, www.cits.ucsb.edu/fake-news/where.

[2] Brownlee, Jason. "A Gentle Introduction to Bayes Theorem for Machine Learning." Machine Learning Mastery, 3 Dec. 2019, machinelearningmastery.com/bayes-theorem-for-machine-learning

[3] Gabielkov, Maksym, et al. "Social Clicks: What and Who Gets Read on Twitter?" ACM SIGMETRICS Performance Evaluation Review, vol. 44, no. 1, 2016, pp. 179–192., doi:10.1145/2964791.2901462.

[4] Ghosh, Souvick, and Chirag Shah. "Toward Automatic Fake News Classification." Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019, doi:10.24251/hicss.2019.273

[5] Pérez-Rosas, Verónica, et al. "Automatic Detection of Fake News." Proc. of the Int'l Conference on Computational Linguistics (2017)., 23 Aug. 2017, arxiv.org/abs/1708.07104.

[6] Shu, Kai, et al. "Fake News Detection on Social Media: A Data Mining Perspective." ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, 2017, pp. 22–36., doi:10.1145/3137597.3137600.

[7] Vosoughi, Soroush, et al. "The Spread of True and False News Online." Science, vol. 359, no. 6380, 2018, pp. 1146–1151.

[8] Zhou, Xinyi, et al. "Fake News: Fundamental Theories, Detection Strategies and Challenge." Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, doi:10.1145/3289600.3291382.

[9] Zhuk, Denis Zhuk, et al. "Methods to Identify Fake News in Social Media Using Machine Learning." *2018 22nd Conference of Open Innovations Association (FRUCT)*, vol. 59, 2018, pp. 401–404.

8. Dataset

[10] Bisaillon, Clément. "Fake and Real News Dataset." Kaggle, 26 Mar. 2020, www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset.