APPROACHES IN FAKE NEWS DETECTION : AN EVALUATION OF NATURAL LANGUAGE PROCESSING AND MACHINE LEARNING TECHNIQUES ON THE REDDIT SOCIAL NETWORK

Moosa Shariff, Brian Thoms, Jason T. Isaacs, Vida Vakilian

Department of Computer Science, California State University, Channel Islands

ABSTRACT

Classifier algorithms are a subfield of data mining and play an integral role in finding patterns and relationships within large datasets. In recent years, fake news detection has become a popular area of data mining for several important reasons, including its negative impact on decision-making and its virality within social networks. In the past, traditional fake news detection has relied primarily on information context, while modern approaches rely on auxiliary information to classify content. Modelling with machine learning and natural language processing can aid in distinguishing between fake and real news. In this research, we mine data from Reddit, the popular online discussion forum and social news aggregator, and measure machine learning classifiers in order to evaluate each algorithm's accuracy in detecting fake news using only a minimal subset of data.

KEYWORDS

Machine Learning, Natural Language Processing, Reddit Social Network

1. INTRODUCTION

Fake news has been considered a significant threat to democracy, journalism, and freedom of expression [1]. In [2], researchers detail the potential for fake news to reduce trust in governments and impact global politics, most notably during the "Brexit" referendum and the 2016 U.S. presidential election. Fake news has also sowed doubt and added additional hurdles when it comes to managing personal health and well-being, as recently experienced from disinformation campaigns surrounding the COVID-19 pandemic and statistics on vaccinations [3]. Fake news can lead to real-world consequences and pose significant challenges to information systems where the goal is delivering relevant, timely and accurate information.

The general population continues to spend more time online each day consuming information, with some estimates that people in the U.S. spend an average of close to 2.5 hours a day on social media [4]. More so, the Internet has emerged as a primary source for entertainment and information that is rapidly replacing traditional media outlets. Consequently, broadcasting features of the Internet, primarily through social media technologies, allow any user to post original content or share content and claim it as 'news'. In many cases, these opinion pieces can often be incomplete information or more deceitful in nature, and in other cases, they can spawn from artificial means such as computer bots. Exacerbating challenges to minimizing fake news is

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 109-124, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120910

the sheer volume and velocity of this information and how quickly it can disseminate within and across social networks. Consequently, as reported in [5], the more exposure a user has to information that is inaccurate or false, the greater the likelihood that they perceived that information as accurate. Additionally, individualistic methods for evaluating fake news range drastically, as reported in [6], and become even more difficult to combat when a person has a personal interest in the story, as reported in [5]. For these reasons, and more, computing solutions are required to minimize exposure to fake news early on and provide users with quick tools for evaluation.

In this research, we focus on Reddit, which has one of the highest percentages of users who receive news, according to Pew Research Center [7], and investigate machine learning models for predicting veracity in Reddit posts using only a minimal subset of data

2. BACKGROUND AND RELATED WORK

2.1. Social Media

According to a report by Elisa Shearer and Jeffrey Gottfried in, "News Use Across Social Media Platforms 2016", an estimated 62% of Americans get news on social media, with around 50% of this population having viewed this news on social media [8]. While interest in news and current events is generally encouraging, access to accurate and reliable information is critical online, where misinformation can be difficult to determine and quick to spread.

Reddit is a web-based platform with features for social news aggregation, content rating, and discussion forums. According to Statista, Reddit has 430 million monthly active users which is slightly higher than the 330 million users of Twitter and it has a higher engagement rate [9]. It had over 199 million posts and 1.7 billion comments in 2019 and is the 5th most visited site in the U.S. More specifically, as reported in [8], 70% of users on Reddit get news from Reddit subreddits (see Figure 1), which is the highest followed by Facebook and Twitter. In other words, individuals flock to Reddit for news, more so than they would to Facebook or Twitter.

Table 1. News by Platform

Platform	Users Receiving News
Reddit	70%
Facebook	66%
Twitter	59%
Tumblr	31%
Instagram	23%
YouTube	21%
LinkedIn	19%
Snapchat	17%
Vine	14%

2.2. Fact-Checking Challenges

Fact-checking is a common technique performed by journalists and involves the verification of claims and sources related to information. With access to a greater number of datapoints than ever before, due in large part to social networking technology and ubiquitous computing, manual fact-

checking is laborious and time-consuming with numerous challenges. Detailed in [10], challenges involve (i) retrieval of all potentially relevant documents, (ii) verification of source reliability, (iii) prediction of source bias and (iv) determination of a document's veracity.

Numerous projects exist today with the goal of impeding the spread of false information online including popular websites. Within the U.S. alone, popular systems such as FactChecker.org, PolitiFact, Snopes and RealClearPolitics. These systems build atop ongoing research in the field, such as work in [11], which uses natural language processing (NLP) to quickly extract and order sentences in ways to aid in the classification of factual claims, and [12] which uses probabilistic classifiers that can both validate credibility but also aid in identifying what aspects of a document a user should focus on. Much other research exists in this emerging area of information systems, but these two studies highlight two critical components of fake news detection, including the challenges involved in data preparation and subsequent steps in algorithm construction, data modelling and testing.

2.3. Natural Language Programming (NLP) Solutions

On Reddit alone, hundreds of thousands of posts are created each day with many posts receiving thousands of views per hour. According to subredditstats.com [13], the top 10 subreddits, i.e. topics, have over 40,000 posts per day and over 360,000 comments each day. Table 2 highlights some of the more popular subreddits and the exposure these topics can generate. Consequently, any viable solution to monitoring veracity in this space would require a system capable of detecting fake news in real-time.

Table 2. Top Subreddit by Post / Comments

Subreddit	**Subscribers**	**Comments**	**Votes**
Politics	7.6m	5.2m	93m
WallStreetBets	10.5m	4.4m	54m
Teenagers	2.5m	880k	52m
NoStupidQuestions	2.3m	720k	9.2m
m=millions, k=thousands			

In this research, we analyze the language used within the titles of Reddit posts. Titles are particularly interesting as they afford Redditors a quick glimpse into a Reddit post and are aimed at attracting viewers with minimal data. Additionally, Reddit post titles are limited to 300 characters. More so, titles typically use language strategically designed to evade detection. Despite this, language leakage occurs, which is hard to monitor. This leakage includes frequencies and patterns of pronoun, conjunction, and negative emotion word usage [14]. The goal in the linguistic approach is to look for such instances of leakage or, so-called "predictive deception cues" found in the content of a message [15]. This can be achieved by creating a machine learning model using NLP algorithms. In this research we use titles to ascertain Ngrams.

N-Grams are sets of keywords that are strung together in groups of *n* words, where *n* is a positive nonzero integer. N-Grams are either continuous sets of characters or words. The most basic version of an N-Gram is the unigram, which is an n-gram of size 1. The next two n-grams are the bigram and the trigram. Frürnkanz [16] noted that word sequences of only about 2 to 3 words were easiest to apply without causing performance stress compared to conducting n-gram analyses on larger word sets. N-Grams can be created from characters, words or even binary text.

In this research, we use a combination of unigrams, bigrams and trigrams as input to our classifiers.

3. RESEARCH METHODOLOGY

3.1. Cross Industry Standard Process for Data Mining (CRISP-DM)

This research adheres to the Cross-Industry Standard Process for Data Mining (CRISP-DM), which is a framework for data mining [17]. Research in fact-checking systems adheres to CRISP-DM in the following steps:

1. Understand the problem domain,
2. Understanding the underlying data,
3. Preprocess and preparing this data,
4. Model the data,
5. Evaluate each model, and
6. Deploy the system.

Understanding the problem domain was covered in 2.1 and 2.2. In this section and subsequent sections, we discuss how we prepare the data, model the data and evaluate each model.

3.2. Data Collection and Pre-processing

Data was collected from two subreddits, theonion and nottheonion. Theonion is a subreddit of satirical Reddit posts and contains links to articles that have fake news. The nottheonion subreddit contains real news. Data was collected from Reddit using Reddit's Python Reddit API Wrapper (PRAW) and Pushshift.io, an API that provides enhanced functionality and search capabilities over PRAW. Initial data collection is agnostic since early in the data mining process we are not concerned with the context of the data, however, limitations in using PRAW are numerous including limiting the result-set from an API call to 1000, preventing the collection of results between specified dates and limiting API calls to only 1 per second. For this reason, we use Pushshift which provides better search functionality and doesn't have any API call limits. Using Pushshift and PRAW 24,001 initial documents were retrieved from nottheonion subreddit and 16,931 initial documents were retrieved from theonion subreddit on January 7, 2020. For all posts the following data was collected:

- Post Title,
- Post Domain from which the article was obtained,
- Number of Comments Per Post,
- Post Timestamp,
- Post Author,
- Post Score, which is calculated using the number of upvotes or downvotes a post received,
- Author's Karma using reddit.info(), which is calculated using a user's contribution to the Reddit community.

After the data collection phases, the pre-processing phase involved cleaning this data for further analysis. Using the Python programming language, more specifically, the numpy and pandas libraries, duplicate entries and stopwords are removed from the dataset. Stopwords were generated using the Python scikit-learn library and a custom algorithm for determining unigrams and bi-grams from the dataset was used. Lemmatization was used instead of stemming to reduce

words to their root form and to help normalize the dataset. Figures 1 and Figure 2 illustrate the Top 5 bigrams for the onion and not the onion subreddits.

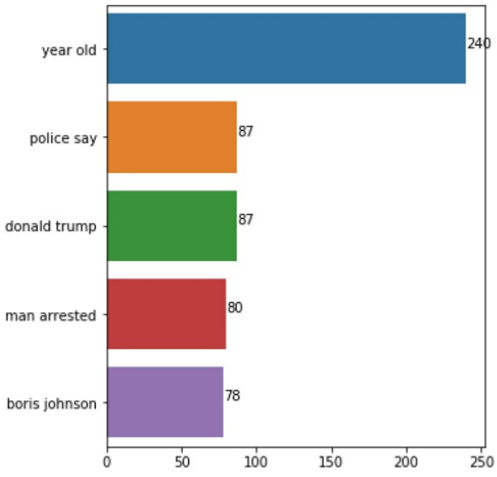

Figure 1. Top bigrams theonion

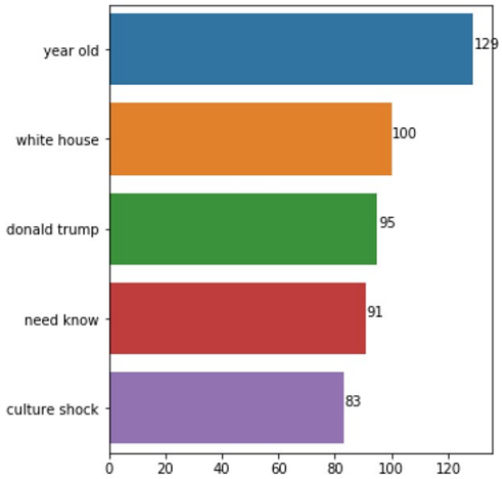

Figure 2. Top bigrams nottheonion

After the data cleansing process, our dataset consisted of 11,201 posts from nottheonion subreddit and 10,605 posts from theonion subreddit. We deemed this to be a good subset since data from each subreddit contained close to the same number of posts. An example title from theonion might look like this: *Neighborhood Rallies To Designate Pothole As Historic Landmark*. An example title from nottheonion might look like this: *Researchers perform magic tricks for birds, who are not amused*.

3.3. Feature Extraction

Feature selection is a critical step in machine learning and allows us to choose those features that contribute the most to the desired output. In other words, which features best afford to classify a post as fake or not. Feature selection not only helps reduce overfitting and improve accuracy but also decreases the training time as there would be lesser data to train. During this stage, we select which parts of the data we want to use from the data we collected. This is done using feature selection algorithms that identify the weight of a feature (i.e. attribute) from our data to affect the classification of our news. Feature selection algorithms reduce the dimensionality of the dataset and select only a minimal subset of features for input into our classifier algorithms. Data is encoded in a numerical format using a Count Vectorizer and TFIDF Vectorizer. A baseline accuracy is calculated prior to parameter hypertuning. KBest and Recursive Feature Elimination from the Python's sklearn feature selection package identify those features which contribute most to our output.

The KBest algorithm provides a score of each feature, where higher scores represent a higher contribution of that feature to the output. Recursive Feature Elimination gives us a ranking of the features by importance and it recursively discards the least important features. Recursive Feature Ranking was also calculated with similar output, which shows lower dependencies on a Reddit post's title and domain name. KBest and Recursive Ranking of feature selection scores can be found in Table 3.

Table 3. Feature Rank

Reddit Feature	KBest Score	Recursive Rank
Author Karma	9458741.764	6
Reddit Score	1874318.802	4
Number of Comments	250962.133	1
Post Title	134261.909	5
Post Author	91729.5	2
Domain	15174.17	3

Table 3 identifies that accuracy scores are significantly determined from a post's karma score and author, followed by the number of comments on the post and then the title of a post. While this feature ranking gives indications to accuracy, they rely on social data affixed to a post after it is already published. Therefore, in this research, we rely primarily on post title and domain for training and testing purposes. As for the authors and their scores, we disregard them because both can be easily spoofed. As detailed in [18], it is estimated that between 9% and 15% of active Twitter accounts were in fact bots and 60 million accounts on Facebook were bots. For Reddit, while numbers are not official, the estimate is around 10% of posts being made by bots.

Finally, features were converted from categorical form to binary form prior to training and testing. The dataset is subsequently split into training and testing datasets (e.g. 80% of the data for training and 20% for testing). After testing, a confusion matrix will be constructed for our best performing model. A confusion matrix helps to describe the performance of a classifier on a set of test data for which the true values are known.

4. CLASSIFIER MODELS AND RESULTS

Classification is the analog of regression when the variable being predicted is discrete, rather than continuous [19]. Classification algorithms are used to predict labels or classes of input data and map data to categorical values. More specifically, a classifier is a function that takes a range of known data as input (i.e. independent variables or predictors) and attempts to group that data into preset categories or classes, which can then be used to classify future unseen data. For each of our models, we focus only on optimized hyperparameters, or those determined through our feature extraction (Section 3.3).

For each model, we calculate three scores. The training score is how well our model performs against the training dataset. The end goal of training is to form a generalized model and prevent overfitting, which occurs when a model fits so well to the data with lots of variance. Validation scores provide insight into average performance over multiple testing iterations. In this research, we focus on n-fold cross-validation, which repeatedly trains on 80% to 90% of our dataset. We set the value of n to 5, based on research in [20], which saw no significant change in the output when increasing the number of cross-validations and a lower number of cross-validations reduces the execution time.

Test Scores are generated once a model has been optimized against relevant hyperparameters suitable for the algorithm. Test scores aim to test unseen data against the validated model and represent how the model would perform in a real-world scenario. Higher scores during testing

indicate a more generalized model. Based on the results during the validation and testing phases, we choose the best performing model in which to refine our final model.

4.1. Model 1: Baseline

Before implementing our classifier models, we use Logistic Regression to obtain a measure for baseline accuracy, our Model 1. Generally, researchers create a baseline model to validate against other models. In statistics, Logistic Regression can be used to model the probability of a certain outcome and is among the top 5 most widely used baseline models [21], largely chosen for its simplicity. For our baseline model, we use Logistic Regression and the title feature from our dataset. The hyperparameters used to formulate the baseline parameters were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 1
- Cost: 1

Scikit-learn's Count Vectorizer can be used to convert a collection of text documents to a vector of term/token counts. For our models, including our baseline model, we use Count Vectorizer to convert Reddit data to a vector of term/token counts. We set the Cost parameter to 1. The cost function for Logistic Regression quantifies the error between predicted values and expected values, which can make the model more complex on one hand but help reduce overfitting on the other hand.

The end accuracy for our baseline model after training, testing and validation was 84.57% and serves as the model to improve upon going forwards. It should be noted that this model showed overfitting with training scores at 99%.

4.2. Model 2 Count Vectorizer and Logistic Regression (1)

Model 2, similar to our baseline model, integrates Count Vectorizer with Logistic Regression but includes more features to train and test (e.g. Title and Domain). Subsequently, two distinct pipelines for our data are generated using a Count Vectorizer on both the Title and the Domain name, to form a single pipeline, which serves as input to our Logistic Regression classifier. The primary parameters and hyperparameters for Model 2 were as follows:

- Features: Title and Domain
- N-gram Range: 1 to 3
- Stopwords: 1
- Cost: 1

Table 4 highlights the results from training and testing and resulted in validation scores over 98%. The high accuracy can be attributed to the fact that domain sources were relatively homogenous, with the majority coming from theonion or a few other sources, all future models omitted domain as a parameter and focus only on the title.

Table 4. Model 2 Output

Measure	Result
Validation Score (%)	98.36
Training Score (%)	99.57
Testing Score (%)	98.27

Elaborating on Model 2, it was decided that we eliminate Domain as a feature for subsequent models. Further analysis of the dataset identified the source for fake news coming largely from two different domains, theonion and clickhole.com. A breakdown of domains and their post breakdown can be found in Table 5 and Table 6.

Table 5. Domains By Reference (Fake)

Domain	Post Count (%)
Theonion.com	7388 (44%)
Clickhole.com	4535 (27%)
Local.theonion.com	1201 (7%)
Politics.theonion.com	1054 (6%)
Youtube.com	457 (3%)
Entertainment.theonion.com	410 (2%)
Sports.theonion.com	382 (2%)
Youtu.be	175 (1%)
Lifestyle.clickhole.com	161 (1%)
i.redd.it	142 (1%)

Table 6. Domains By Reference (Real)

Domain	Post Count (%)
Theguardian.com	727 (3%)
Cnn.com	689 (3%)
Foxnews.com	582 (2%)
Google.com	565 (2%)
Bbc.com	561 (2%)
Independent.co.uk	474 (2%)
Nbcnews.com	471 (2%)
Nypost.com	439 (2%)
Newsweek.com	434 (2%)
Bbc.co.uk	381 (1%)

4.3. Model 3 Count Vectorizer and Logistic Regression (2)

For Model 3, we eliminate the Domain feature from our model and rerun our testing using Count Vectorizer and Logistic Regression. This creates a single pipeline for data input. Results are detailed in Table 4 and show validation scores around 84.94%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 1
- LRC: 1

Table 7. Model 3 Output

Measure	Result
Validation Score (%)	84.94
Training Score (%)	99.84
Testing Score (%)	84.62

4.4. Model 4 TF-IDF Vectorizer and Logistic Regression

Model 4 integrates TF-IDF with Logistic Regression. Term frequency/inverse document frequency (TF-IDF) is one of the most commonly used term weighting schemes in today's information retrieval systems [22]. This is different from Count Vectorizer as it takes into account the occurrence of the word not just in a single document but in the entire set of documents. Common words like 'a', 'the', etc. that appear frequently across all documents, have reduced weights and more weightage is given to words with lower frequency counts. To convert a collection of raw documents to a matrix of TF-IDF features we find the product of term frequency and inverse document frequency.

Results for Model 4 are detailed in Table 8 and show validation scores of 84.04%. The primary parameters and hyperparameters used in this model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Cost: 1

Table 8. Model 4 Output

Measure	Result
Validation Score (%)	84.04
Training Score (%)	91.84
Testing Score (%)	84.00

4.5. Model 5 Count Vectorizer with Support Vector Machine (SVM)

For Model 5, we implement a Support Vector Machine (SVM). SVM is a supervised machine learning algorithm that is capable of performing linear and nonlinear classification. SVM is an optimal classifier in the sense that, given training data, it learns a classification hyperplane in the feature space which has the maximal distance (or margin) to all the training examples, with the exception of a small number of outlier examples [23]. Linear kernels such as SVM are preferred for text classification due to a number of reasons including that most text classification problems are linearly separable [24] and text tends to possess many features, which is good for a linear kernel where non-linear mapping does not improve the performance [25]. Results for Model 5 are detailed in Table 9 and show validation scores at 84.85%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Cost: 0.1

Table 9. Model 5 Output

Measure	Result
Validation Score (%)	84.85
Training Score (%)	99.88
Testing Score (%)	85.50

To elaborate more on Model 5, SVM costs are calculated differently from Logistic Regression, thus the difference in these hyperparameter values. However, Cost is used in a similar fashion to Logistic Regression and attempts to guide how much we want to avoid misclassifying the data. Higher cost values help avoid misclassifying data but increase execution time.

4.6. Model 6 Count Vectorizer with Random Forest Classifier

Random forest classifier is a supervised machine learning algorithm based on ensemble learning. In contrast to ordinary learning approaches which try to construct one learner from training data, ensemble methods try to construct a set of learners and combine them [26]. According to [27] there have been significant improvements in classification accuracy by growing an ensemble of trees and letting them vote for the most popular class. Although it generally gets a better accuracy, this is not true for all predictions and it is slow to generate predictions when multiple decision trees. Results for Model 6 are detailed in Table 10 and show validation scores at 81.27%. It should be noted that Model 6 was the lowest-performing model. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Max Depth: None
- Min Samples Leaf: 1
- Min Samples Split: 5
- Estimators: 200

Table 10. Model 6 Output

Measure	Result
Validation Score (%)	81.27
Training Score (%)	99.92
Testing Score (%)	81.85

4.7. Model 7 TF-IDF Vectorizer with Multinomial Naïve Bayes

Model 7 implements a Naïve Bayes Classifier. Naïve Bayes Classifiers are based on applying Bayes Theorem with the “naïve” assumption of conditional independence between features. Recent work in supervised learning has shown that a surprisingly simple Bayesian classifier with strong assumptions of independence among features, called Naïve Bayes, is competitive with state-of-the-art classifiers such as C4.5 [28]. Naïve Bayes classifiers use Bayes Theorem, which calculates the probability of an event based on the prior knowledge of conditions that might be related to the event. While Naïve bayes classifiers are simple, they can provide are fast and accurate.

The Multinomial Naïve Bayes classifier is a specialized version of Naïve Bayes that is suitable for classification with discrete features (e.g., word counts for text classification). Multinomial Naïve Bayes estimates the conditional probability of a particular term given a class as the relative frequency of the term t in all documents belonging to the class C [29]. It assumes that every word is independent of the other. Instead of calculating the probability of sentences, we now calculate the probability of every single word. These probabilities are multiplied and the highest probability gives us the class it belongs to. Results for Model 6 are detailed in Table 11 and show validation scores at 83.94%. The primary parameters and hyperparameters defined for our baseline model were as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Alpha: [0.1, 0.3, 0.6, 1]

Table 11. Model 7 Output

Measure	Result
Validation Score (%)	83.94
Training Score (%)	90.53
Testing Score (%)	83.51

4.8. Model 8 Count Vectorizer with Multinomial Naïve Bayes

In Model 8, we implement features of Model 3 and Model 7 using Multinomial Naïve Bayes Classifier with Count Vectorizer. Results for Model 8 are detailed in Table 12 and show validation scores at 85.17%. It should be noted that Model 8 was the highest performing model. The primary parameters and hyperparameters defined for our baseline model are as follows:

- Feature(s): Title
- N-gram Range: 1 to 3
- Stopwords: 0
- Alpha: [0.1, 0.3, 0.6, 1]

Table 12. Model 8 Output

Measure	Result
Validation Score (%)	85.17
Training Score (%)	99.53
Testing Score (%)	85.50

Since this model generated the highest performance, we select this model as ‘Best’ and conduct a confusion matrix to further determine the model’s accuracy. Illustrated in Figure 3, the confusion matrix labeled 2301 true negatives, 350 false positives, 440 false negatives and 2361 true positives. 790 predictions were misclassified. Using these values, we calculate accuracy, precision and recall for this model. This results in an accuracy of 85.51%, precision of 87.09%, recall of 84.29% and F1-score of 85.67%. The accuracy for this model is better than our baseline accuracy.

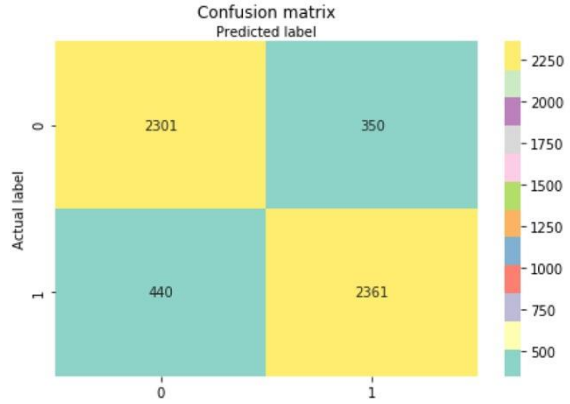

Figure 3. Model 8 Confusion Matrix

5. DISCUSSION

In this section, we reflect on the implications of our results and identify a few key findings surrounding natural language approaches in fake news detection.

5.1. Feature Removal - Domain

Performing KBest and Recursive Feature Ranking allowed us to produce a subset of features to serve as input for our model. It is important to note that most features were omitted since much of the data comprised in those features, such as comments and postscores are values that are generated after a post is submitted. Instead, we rely solely on values that are generated at the time of the post. Furthermore, results from Model 2 indicated a high degree of fitting using Domain plus Title. However, further analysis identified Domain to be a poor feature since it was relatively homogenous and varied little across theonion subreddit. This is most likely due to the fact that most of our data belongs to two to three domains, most of the domains referenced in theonion are from theonion.com (7388), clickhole.com (4535) and the rest of them are referenced a very few number of times. This allowed us to eliminate Domain as a parameter and focus on a complete minimal subset of our data and only use Title. Consequently, computing Logistic Regression using title and Count Vectorizer resulted in an accuracy of 84.57%.

5.2. NLP Considerations

Our research supports the use of n-grams in fact-checking systems, which has shown particular success in previous studies, such as [30], which used trigrams and gradient boosting to achieve 95% accuracy on an open-source Kaggle dataset. Using n-grams helped the classifiers capture complex expressions which in turn helped increase their accuracy. None of the N-grams and stop words from the NLTK package were selected. Additionally, we discovered that models not relying on lemmatization achieved slightly higher performance early on. Therefore, it was decided to proceed using non-lemmatized words.

5.3. Over Fitting

Overfitting takes place when a model achieves high testing scores with low validation scores. Models tend to overfit when a dataset is filled with noise and/or inaccurate data. One solution to avoid overfitting is to use linear algorithms on linear data or setting hyperparameters such as maximum depth if we are using decision trees.

Disregarding our baseline model, overfitting tended to take place in Model 3, Model 5, Model 6 and Model 7. Overfitting is more common in linear models such as Logistic Regression and Support Vector Machines. It was surprising to find Model 7 overfitting since Random Forest classifiers are generally better at preventing overfitting, but this was not the case. More so, our best-performing algorithm also showed overfitting in its training and results. It would be worthwhile to explore different approaches for Model 4 and Model 8, which were the only two models not to experience overfitting.

5.4. Hyper Parameters

Hyperparameter optimization or tuning is a critical aspect of machine learning. A hyperparameter is a parameter whose value is used to control the learning process. Each model required different hyper-tuning parameters depending on the algorithm and its requirements. For example, the alpha hyperparameter for our Multinomial Naïve Bayes classifiers differed from Model 5 (0.6) and Model 6 (1.0). The alpha parameter, also known as the additive smoothing parameter controls the shape of our model. A lower score makes the model complex, whereas a higher score makes the model simple and biased. Additive smoothing adds to the probability and is used as a fail-safe for unknown words in the vocabulary.

Hyper-tuning variables for each of the classifiers achieved only slight improvements of the test scores (e.g. 0.05% to 0.1%). All models used a cross-validation of 5 which repeatedly trains the model on the dataset 5 times.

5.5. Count Vectorizer vs. TF-IDF

It was interesting to note that Count Vectorizer performed slightly better than TF-IDF. This is typical since TF-IDF generally performs better on larger datasets. TF-IDF reflects the relative importance of a word for statistical analysis and is generally better than Count Vectorizers because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words. Unfortunately, TF-IDF was only performed with Logistic Regression and multinomial naïve Bayes classifier. In future research, it also makes sense to extend the implementation to support vector and random forest classifiers. The implementation of TF-IDF across other machine learning approaches may result in better results.

6. LIMITATIONS AND FUTURE RESEARCH

The authors acknowledge that a number of limitations in this research exist. First, data collected was from a single day in January of 2020. Future research should look to expand the data collection phase. Next, improvement to each of the models could be made by using bootstrap aggregation, also called bagging and/or boosting which has shown to reduce bias and improve the stability and accuracy in statistical classification and help prevent overfitting. Also, models applying TF-IDF algorithms appeared to resist overfitting, so it would be interesting to apply TF-IDF with Support Vector Machine and Random Forest. Finally, each model could be expanded to collect text from within an article. This might better train each model. In this research, we focus solely on Title since it is can play a critical factor in determining whether or not a user chooses to view a post.

7. CONCLUSION AND FUTURE WORK

In this research, we implement and evaluate multiple classifier models, which can be used to aid in fake news detection, all of which performed well. The best model was built using Count Vectorizer and Multinomial Naïve Bayes and was able to get an accuracy score of 85.51%. While accuracy levels are low, it is close to the accuracies obtained by [31] and [32] on social media platforms. Interestingly in this research, we were able to get an accuracy above the baseline accuracy by the implementation of different machine learning classifiers and through hyper-tuning. Additionally, unlike the other statistical models on fake news, which use a variety of metadata from social media platforms, our findings rely only on Reddit post titles. This work demonstrates that titles can play a significant role in classifying information as real or fake and marks a good starting point for detecting fake news.

REFERENCES

[1] Frearson, J. (2018). "The rise of fake news is a threat to our democracy - and our message," *Business Reporter*, Jan. 2018. Retrieved online via: https://www:businessreporter:co:uk/2018/01/13/rise-of-fake-news-is-a-threat-to-our-democracy.

[2] Zhou, X. and Zafarani, R. (2018). "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys* (CSUR), 2018.

[3] van der Linden, S. Roozenbeek, J. and Compton, J. (2020). "Inoculating Against Fake News About COVID-19," *Frontiers in Psychology*, v11), 2020.

[4] Tankovska, H. (2021). "Daily time spent on social networking by internet users worldwide from 2012 to 2020," Statista. Originally published on Feb 8, 2021. Retrieved online on https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/.

[5] Balmas, M. (2014). "When Fake News Becomes Real: Combined Exposure to Multiple News Sources and Political Attitudes of Inefficacy, Alienation, and Cynicism," *Communication Research*, v41(3), pp. 430–454.

[6] Zafarani, R., Zhou, X., Shu, K. and Liu, H. (2019). "Fake News Research: Theories, Detection Strategies, and Open Problems," In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (KDD '19), Association for Computing Machinery, New York, NY, USA, pp. 3207–3208.

[7] Shearer, E. and Mitchell A. (2021). "News Use Across Social Media Platforms in 2020," Pew Research Center, Published January 12, 2021.

[8] Shearer E. and Gottfried, J. (2016). "News use across social media platforms. News use across social media platforms," *Pew Research Center*, May 2016. Retrieved from https://www:journalism:org/2016/05/26/news-use-across-social-media-platforms-2016/.

[9] Tankovska, H. (2021). "Global social networks ranked by number of users 2021," Statista. Originally published on Feb 9, 2021. Retrieved online on https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[10] Nadeem, M., Fang, W. Xu, B. Mohtarami, M. and Glass, J (2019). "Fakta: An automatic end-toend fact checking system," In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[11] Hassan, N., Arslan, F., Li, C., and Tremayne, M. (2017). "Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster," In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '17). Association for Computing Machinery, New York, NY, USA, pp. 1803–1812.

[12] Nguyen, TT, Weidlich, M., Yin, H., Zheng, B., Nguyen, QVH, and Stantic, B. (2019). "User guidance for efficient fact checking," In *Proceedings of the VLDB Endowment*. v12(8). April 2019, pp. 850–863.

[13] Reddit.com, "Subreddit Status," Retrieved online from https://subredditstats.com/ on June 9, 2021.

[14] Feng, VW and Hirst, G. (2013). Detecting deceptive opinions with profile compatibility. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 338346, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

[15] Conroy, NK, Rubin, VL, and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news," In *Proceedings of the Association for Information Science and Technology*, 52(1), January 2015.

[16] Frürnkanz, J. (1998). "A Study Using N-Gram Features for Text Categorization," *Austrian Research Institute for Artificial Intelligence*, pp. 98-30, 1998.

[17] Shearer C. (2000). "The CRISP-DM model: the new blueprint for data mining," *Journal of Data Warehousing*, v5, pp. 13-22.

[18] Lazer, DMJ, Baum, MA, Benkler, Y., Berinsky. AJ, Greenhill, KM, Menczer, F, Metzger, MJ,
Nyhan, B, Pennycook, G, Rothschild, D., Schudson, M., Sloman, SA, Sunstein, CR, Thorson, EA, Watts, DJ, Zittrain, JL (2018). "The Science of Fake News," *Science* , March 09, 2018, v359(6380), pp. 1094-1096.

[19] Pereira, F, Mitchell, T., and Botvinick, M. (2009). Machine learning classiers and fMRI: A tutorial overview. NeuroImage, 45(1): S199{S209, March 2009.

[20] Forman, G. and Cohen, I. (2004). "Learning from little: Comparison of classifiers given little training," In *Lecture Notes in Computer Science*, pp. 161-172. Springer Berlin Heidelberg,.

[21] Lin, W., Hu, Y. and Tsai, C. (2012). "Machine learning in financial crisis prediction: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v42(4), pp. 421-436.

[22] Aizawa, A. (2003). "An information-theoretic perspective of tf-idf measures," *Information Processing and Management*, v39(1), pp. 45-65, January 2003.

[23] Li, Y., Bontcheva, K. and Cunnigham, H. (2009). "Adapting SVM for data sparseness and imbalance: a case study in information extraction," *Natural Language Engineering*, v15(2), pp. 241271, April 2009.

[24] Joachims, T. (1998). "Text categorization with support vector machines: Learning with many relevant features," In *Machine Learning*: ECML-98, pp. 137-142, Berlin, Heidelberg, 1998.

[25] Hsu, CW, Chang, CC and Lin, CJ. (2008) "A practical guide to support vector classification," *Technical Report Department of Computer Science and Information Engineering*, National Taiwan University.

[26] Zhou, ZH. (2012). "Ensemble Methods: Foundations and Algorithms," *Chapman and Hall/CRC*, 1st Edition.

[27] Breiman, L. (2001). "Random forests," *Machine Learning*, v45(1), pp. 5-32.

[28] Friedman, N., Geiger, D. and Goldszmidt, M. (1997). "Bayesian network classifiers," *Machine Learning*, v29(2/3), pp. 131-163.

[29] Kamel S. (2019). "Arabic Language Processing: From Theory to Practice," *Springer International Publishing*.

[30] Wynne, HE and Wint, ZZ. (2019). "Content Based Fake News Detection Using N-Gram Models," In *Proceedings of the 21st International Conference on Information Integration and Webbased Applications & Services* (iiWAS2019). Association for Computing Machinery, New York, NY, USA, pp. 669–673.

[31] Ajao, O, Bhowmik, D. and Zargari, S. (2018). "Fake news identification on twitter with hybrid cnn and rnn models," In *Proceedings of the 9th International Conference on Social Media and Society*, SMSociety '18. Association for Computing Machinery, pp. 226-230, New York, NY, USA.

[32] Ruchansky, N., Seo, S. and Liu, Y. (2017). "Csi: A hybrid deep model for fake news detection," In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, Association for Computing Machinery, pp. 797-806, New York, NY, USA, 2017.

AUTHORS

Moosa Ahmed Shariff is a recent graduate of the Master of Computer Science program at California State University Channel Islands. His research interests explore the implementation of machine learning algorithms on large data sets using python and visualization software. After graduating CSU Channel Islands, his career goals are to acquire a software engineering position with an emphasis on data science.

Dr. Brian Thoms is Associate Professor of Computer Science at California State University, Channel Islands where he teaches courses on Human Computer Interaction and Database systems. Dr. Thoms received his PhD in Information Systems and Technology from Claremont Graduate University. His research interests explore data mining in the domains related to social and healthcare systems. He is also co-founder and CIO for Health e-Services, a health storage and analytics company.

Dr. Jason T. Isaacs is Associate Professor of Computer Science at California State University, Channel Islands where he teaches courses on embedded systems and software engineering. Dr. Isaacs received his Ph.D. degree in Electrical and Computer Engineering from the University of California, Santa Barbara. His research interests include multiagent control systems, UAV path planning, localization and mapping, and sensor networks.

Dr. Vida Vakilian is an Assistant Professor of Computer Science at California State University, Channel Islands. From 2015 to 2019, she was an Assistant Professor at California State University, Bakersfield. Dr. Vakilian received her Ph.D. degree in Electrical Engineering from University of Montreal, Canada in 2014. Her research interests include wireless communication, signal processing and communication-aware robotics. Dr. Vakilian is the recipient of the Enhancing Access to the Radio Spectrum (EARS) Award from the NSF.