World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:17, No:3, 2023

Sentiment Analysis of Fake Health News Using Naive Bayes Classification Models

Danielle Shackley, Yetunde Folajimi

Open Science Index, Cognitive and Language Sciences Vol:17, No:3, 2023 publications.waset.org/10012995.pdf

Abstract—As more people turn to the internet seeking health related information, there is more risk of finding false, inaccurate, or dangerous information. Sentiment analysis is a natural language processing technique that assigns polarity scores of text, ranging from positive, neutral and negative. In this research, we evaluate the weight of a sentiment analysis feature added to fake health news classification models. The dataset consists of existing reliably labeled health article headlines that were supplemented with health information collected about COVID-19 from social media sources. We started with data preprocessing, tested out various vectorization methods such as Count and TFIDF vectorization. We implemented 3 Naive Bayes classifier models, including Bernoulli, Multinomial and Complement. To test the weight of the sentiment analysis feature on the dataset, we created benchmark Naive Bayes classification models without sentiment analysis, and those same models were reproduced and the feature was added. We evaluated using the precision and accuracy scores. The Bernoulli initial model performed with 90% precision and 75.2% accuracy, while the model supplemented with sentiment labels performed with 90.4% precision and stayed constant at 75.2% accuracy. Our results show that the addition of sentiment analysis did not improve model precision by a wide margin; while there was no evidence of improvement in accuracy, we had a 1.9% improvement margin of the precision score with the Complement model. Future expansion of this work could include replicating the experiment process, and substituting the Naive Bayes for a deep learning neural network model

Keywords—Sentiment analysis, Naive Bayes model, natural language processing, topic analysis, fake health news classification model.

I. Introduction

FAKE health news classification models were initially developed after the propagation of online news increased. This is a machine learning classification problem and is investigated through the use of different models and features. Amidst the COVID-19 pandemic, the internet has become a critical part of peoples' daily lives. In the heat of the pandemic, local governments were enforcing lockdowns to curb the spikes in COVID-19 cases. This led to huge switches to the internet for nontraditional reasons. Activities like grocery shopping and attending workout classes can now be accomplished without leaving the house. Working from home has also become the new norm for millions. A significant percentage of workers have been impacted. In 2019, fewer than 6% of Americans worked from home. In May 2020, (the beginning of the pandemic) over 48 million people were working from home (around 35% of the employed work force) [1]. Consequently, the public turned to online health resources, searching for doctors' offices, treatments and symptom searching. As people were trying to research, learn more about the pandemic and how to keep safe, they were being inundated with fake health information. False health information has been plaguing the population for years. One of the most spread health falsities was the misconception that the measles, mumps and rubella vaccine causes autism. This rumor preyed on parents of toddlers for fear that their child could develop autism if they were vaccinated.

The concerns of trust in artificial intelligence (AI) with medical diagnosis can be summed up from the surveys done by Juravle et al. [2]. They investigated trust in AI as a medical diagnosis tool using a user study. Their findings "highlight that people have comparable standards of performance for AI and human doctors and that trust in AI does not increase when people are told the AI outperforms the human doctor". This means that users expected the same performance from a human doctor and AI. Even when AI outperformed a human doctor, the users did not trust the AI more. However, the gap in trust between AI and human doctors is lessened when users were able to choose their own doctor, the human or AI. This highlights a potential acceptance of AI in the future. A step towards the public trusting AI is aiding the development of tools to keep people safe from health misinformation. While there are existing models using sentiment analysis to address these kinds of problems on fake news classification models, there is a lack of conclusions about the effects of sentiment analysis specifically on fake health news.

Fake health news is defined as "fabricated information that mimics news media content in form but not in organizational process or intent" [3]. It can also be characterized by its "deliberate reporting of lies or misleading interpretation of facts" [3]. To further corroborate how prevalent fake health information is, a 2020 study done by the Independent [3] mentions that, "Of the 20 most-shared articles on Facebook with the word 'cancer' in the headline, more than half report claims discredited by doctors and health authorities". Social media provides platforms for fake articles to be accessed, interacted with and shared more frequently. While there are multiple successful models for classifying fake news, there is a lack of fake health news classification models. While copious amounts of fake and real health data exist online, a classification model needs credibly labeled data. This gap of research between fake news and fake health news classification models is attributed to the overall sparsity of data of fake health news.

Natural language processing (NLP) is defined as the "joint field of computer science, artificial intelligence, and

Danielle Shackley and Yetunde Folajimi are with the School of Computing and Data Science Department, Wentworth Institute of Technology, Boston, MA, 02115 USA (e-mail: dshackley25@gmail.com, folajimiy@wit.edu).

linguistics that focuses on the interaction between machine and human" in [4]. The purpose of this field is to "achieve effective and efficient communication between human and machine through natural language" [4]. NLP is comprised of three main phases, data preprocessing, algorithm development and evaluation. There are various applications of methods available for implementation, such as sentiment analysis, text summarizations and topic analysis. Before the algorithms can be implemented, there is extensive preprocessing that needs to convert data into usable format for a computer, called word embedding. Some challenges of NLP include the ambiguity of natural language. Words can have several meanings and sentences can have multiple meanings, depending on context. There are tools specifically created for NLP problems, the Natural Language Toolkit (NLTK) [5] is a collection of Python libraries that perform various jobs within an NLP application.

This research uses a subfield of NLP, called text classification. Text classification is a supervised machine learning technique, that given data, trains based on an algorithm or model to predicts a label Its applications include spam filtering, email routing and sentiment analysis [6]. The use of text classification in this work is taking fake and real health text data and training a model to be able to predict the label correctly.

Naive Bayes classification models are implemented in this work. Naive Bayes classification models are based off of Bayes Theorem. They are commonly used models in text classification problems. Their simplicity is one of the biggest draws to this type of model. Naive Bayes models work off the assumption that all the values of a particular feature are independent of the value of any other feature. The models assume the effect of the value of a predictor (x) on a given class (c) is independent of other predictors, this is referred to as conditional independence [7].

This paper explores the use of machine learning for classifying fake health news and NLP techniques. We include relevant literature reviews that give background on topics implemented in our model. The data preprocessing steps, model implementation, and sentiment analysis results are explored. We show a comparison of model implementation performances, as well as trends found within the fake and real classified datasets. Lastly, the conclusion summarizes the findings of this work as well as proposes future work.

II. Related Works

The work done by Sameer et al. [8] shows the performance of an analysis on features associated with reliable and unreliable media sources. Their experiment involved the development of a health-oriented news dataset with both unreliable and reliable sources with over 30,000 health related news articles dated between 2015-2018. Using their dataset to perform systematic content analysis, they identified structural, topical and semantic differences between the two types of information sources. Structural analysis found that unreliable media outlets use clickbait sounding headlines more often than reliable sources. Topical results showed that when reliable sources discuss 'cancer' there are other research and studies mentioned. Unreliable outlets associated 'cancer' with autism and vaccination. Lastly, semantic analysis showed average health news from reliable sources contained more references and quotes than unreliable. To prove the efficiency of the analysis in their contribution towards classification of real or fake health news, the authors developed a machine learning model that can predict an article with an F-measure of 96%.

Another important research topic highlighted by Hakak et al. [9] proves the advantages of effective feature extraction on classification of fake news. The experiment took two large datasets for news classification, ISOT and the Liar dataset. The data are cleaned and feature extraction is performed on 26 features in an ensemble approach. Three machine learning models, Decision Tree classifier, Random Forest Tree algorithm, and Extra Tree classifier are used in a bagging approach to aggregate the output of the models. After, the datasets were split into training and testing sets using k-fold validation. Hyperparameters are adjusted with a random search method to determine the optimal value for these default parameters. Feature extraction is an important part of classification problems because it reduces the dimensionality of the data, eliminating irrelevant features and helping improve accuracy. Their experiment results on the ISOT dataset of 99.8% and 44.15% on training and testing respectively and 100% on the Liar dataset proves this methodology is important to consider in future models.

Naive Bayes is a very common model applied to spam filtering problems. Granik et al. [7] wanted to prove that Naive Bayes can be used to classify fake news even on a small dataset. Fake news articles often contain the same sets of words, thus Naive Bayes classifiers are a good suit for this type of application. The appearance of known associated fake words affect the probability of a text being fake. They mention the similarities in patterns between spam messages and fake news. These include grammatical mistakes and emotionally colored text. Their content often tries to affect the reader's opinion in a manipulative fashion and often use similar sets of words. Assuming there is a training set with labeled data as true and fake, the probability of finding a specific word in fake news articles can be defined as a ratio of fake news articles that contain the word to the total number of fake news articles. The same is true for real news articles. The dataset used in this experiment was collected and hand labeled as "mostly true", "mostly false", "mixture of true and false" and "no factual content" by Buzzfeed reviewers. The preprocessing on these data discarded any samples that had missing fields, and any content labeled "mixture of true and false" or "no factual content". After preprocessing, the number of posts totaled to 1,771 samples. The data were shuffled and split into training, testing and validation sets. After training the model, the testing set had accuracies on the real and fake data of 75.59% and 71.73% respectively. Their recommendations of improving the model included adding training data, removing stop words and using the NLP technique of stemming.

The Valence Aware Dictionary and sEntiment Reasoner (VADER) was tested on Tweets collected about the 2016 presidential election in order to classify sentiment in [10]. They used a multi-classification system to classify Tweets as either highly positive, positive, neutral, negative or highly negative, mentioning that in most other studies only a binary classification system was used. Twitter has become a rich source of data that are highly used in fields of opinion or sentiment analysis. Their preprocessing steps consisted of removing stop words, converting to lower case, tokenizing and stemming with the Porter Stemmer. They collected 2,000 Tweets for this study by using Network Overview Discovery and Exploration for Excel. They used date constraints to ensure that the topic was only the 2016 election. Since VADER does not require training data, the researchers examined the sentiment polarity percentage for Tweets within a category as well as the count for Tweets within a category. Some of the classified positive words included ‘win’, ‘support’, ‘like’, while negative contained ‘protest’ and ‘riot’. Their results showed that this is a successful tool in a multi-class sentiment analysis system and that one of VADERS biggest advantage is its ability to quickly classify huge amounts of data.

III. Methodology

The objective of this research included collecting sufficient reliably labeled health information, implementing Naive Bayes models and evaluating the performance of the models based on the inclusion of a sentiment analysis feature. Lastly, we use LDA topic analysis models to identify potential explanations of the impact we saw with the addition of sentiment analysis.

A. Data Collection

There are abundant sources of data for fake news models, as it has been a research topic in the computer science field for years. However, the lack of fake health news data was addressed by the work done in [11].

The data repository *FakeHealth* [12] published on GitHub, contains the first comprehensive fake health news dataset. The collection contains a total of 2,296 articles. It is broken into two different subsections: HealthStory and HealthRelease. HealthStory contains news stories that are posted by news media. HealthRelease contains news releases from institutions, universities, research centers and companies. Within both of these subsections, the data can be categorized as one of four topics: news content, news reviews, social engagements and user networks. The features included with this dataset include url, title, key words, text, images, tags, authors, date, rating, ground truth of rating criteria, explanations of the ground truth, tags, category, title, summary, description, images. and news source. In this investigation, we only use the title and rating.

The ‘rating’ feature comes from HealthNewsReviews own criteria for rating news sources. They have compiled a list of criteria questions such as “Does the story adequately discuss the costs of the intervention” and “Does the story adequately quantify the benefits of the intervention”. Each question is an evaluation of the credibility and evidence of each article in order to fully evaluate. Each of the criteria questions are assigned values of “Satisfactory,” “Unsatisfactory” or “Not Applicable” and these are converted into scores. The 10 values from the criteria are converted into percent of criteria judged satisfactory and given a percentage score. These percentages are converted into “star” ratings.

TABLE I Title from FakeHealth Dataset

Text	Label
Google retinal scans can predict if you will have a heart attack.	Fake
Experimental blood test could detect melanoma skin cancer early study finds.	Real

The data repository contained 2,296 rows of data. Examples from the dataset are in Table I.

B. Data Preprocessing

In order to use text data, the data were preprocessed after the collection phase. Fig. 1 shows the flow of the data preprocessing phases. The first step was removing all non English records. Next, all non alpha numeric characters were removed. For the purposes of classifying fake and real health news, it was decided that these characters would not benefit the model. We converted all text to lowercase and then split each text into a list. The Porter Stemmer [13] was used to stem each word within each list. The stemming technique is used to reduce inflectional and derivational related forms of a word and convert the word to its base form. Stemming refers to the heuristic process of cutting off the ends of words, leaving the common base form [14].

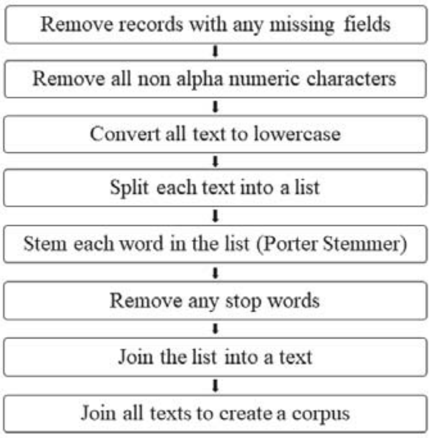
Remove records with any missing fields
Remove all non alpha numeric characters
Convert all text to lowercase
Split each text into a list
Stem each word in the list (Porter Stemmer)
Remove any stop words
Join the list into a text
Join all texts to create a corpus

Fig. 1 Data Preprocessing Steps

As the texts were stemmed, the NLTK *english* stop words corpus was used to filter out stop words. The removal of stop words is a commonly used method that is used to improve text models. Stop words can be defined as words that, “safely be ignored without sacrificing the meaning of the sentence” [15]. Words are removed because they are invaluable to the model and can even be detrimental because of their frequency. Common stop words include: the, is, at, which, and on. There are different collections of stop words that can be implemented depending on the task. Lastly, the words were joined together into a text, and all of the texts were joined to create a corpus.

C. Sentiment Analysis

To generate the sentiment analysis of the text data, the VADER model was implemented. VADER is a sentiment analysis model that is sensitive to both positive and negative sentiment as well as the strength of emotion. VADER is available from the NLTK package. The model relies on a labeled dictionary that maps lexical features and words to an emotion. The sentiment scores of a text are generated by summing the intensity of each word's emotion within a text [16], [17]. In this work, we generate the sentiment analysis of each text in the dataset. VADER takes in a text and returns a dictionary of scores including negative, neutral, positive, and compound. The compound score is computed by normalizing the negative, neutral and positive scores. This compound score is the used to classify the sentiment

VADER was chosen because it can be implemented on unlabeled data. VADER is simple to instantiate, by importing the *SentimentIntensityAnalyzer* from the NLTK package, the scores can be generated by looping through all of the text to be analyzed. For this purpose, since only the compound scores were going to be used to label the sentiment, we converted the dictionary output of each score to just the compound score. As each text was given a compound score, the value was checked for the associated sentiment from the metrics defined above, and the negative, neutral or positive sentiment was assigned. An example of a labeled sentiment score can be found in Table II.

TABLE II of Data Classified by the VADER Sentiment Analyzer

Classified Sentiment	Text
Positive	New findings could save lives of more stroke patients
Negative	In Italy 232 children have died from COVID-19

The outputs of VADER sentiment analysis on the dataset included 1,222 negative, 1,337 positive and 439 neutral. The distribution of real and fake data classified are in Table III. After generating the sentiment analysis on all records in this dataset we can see that there is a distributed amount of each sentiment and of real and fake.

TABLE III
SUMMARY OF RESULTS FROM VADER SENTIMENT ANALYSIS

	Real	Fake	Total
Positive	593	744	1337
Neutral	215	224	439
Negative	494	728	1222
Total	1302	1696	

D. Models

A total of 6 classification models were created, parameter tuned and compared. To test the additional of the sentiment analysis feature, the same Naive Bayes Classifier models are implemented with and without the feature. The basics of each model use the text, a custom character and token counter feature. The sentiment analysis is added to the existing models to test for improvement, this way the sentiment analysis is the only factor affecting the models.

Frequency models were created to visually show the counts of the most common words within a text. Below are graphs with the top 15 most common words that appear in the fake and real datasets. They are separated by their classification to highlight the similarity in their contents, which provides insight into the difficulty of a classification task on this type of data. Stop words using the NLTK *english* import were removed as well as the words 'say', 'amp', 'may' and 'covid'. We decided to remove 'covid' because it was the top word for both datasets and we wanted to analyze other patterns.

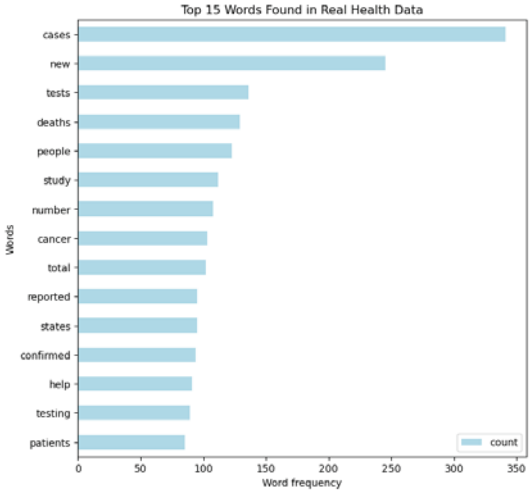

Fig. 2 Real Health Data Top 15 Most Frequent Words

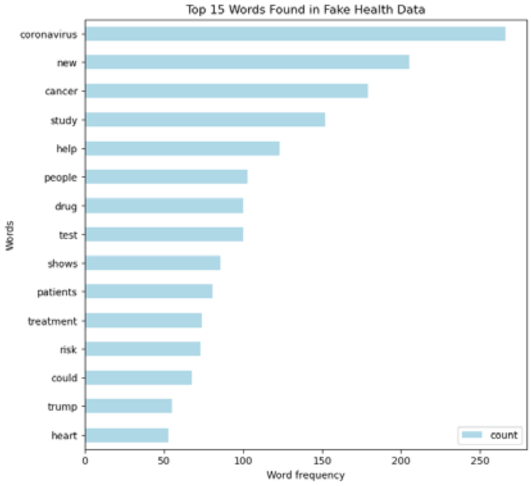

Fig. 3 Fake Health Data Top 15 Most Frequent Words

The most frequently occurring words in the real and fake health sets were 'coronavirus' and 'cases' respectively. The top common words for both sets were extremely similar. They both have 'new', 'tests', 'people', 'study', 'cancer', 'help' and 'patients' amongst the most common. The real dataset saw less frequent usage of the word 'cancer', as it was the 8th most common, the fake set had it as the 3rd most common. Lastly, the real dataset contained 'reported' and 'confirmed' while the fake data had neither of these.

The character and token counter methods are responsible for finding the length of a text and counting the number of alpha numeric tokens within each text. In natural language processing problems, patterns in the lengths of texts can be a useful feature, so we included these in the models.

Vectorizers are needed to transform the textual data into numerical vectors for the model to train with. We decided to experiment with both CountVectorizer and TFIDF vectorizer. The TFIDF vectorizer uses the statistic based on the frequency of a word within a corpus, and provides a numerical value representing its importance [18]. Some works such as [19] have found the TFIDF vectorizer as the preferred choice in classification problems. However, in this project we found the CountVectorizer provided better classification results. CountVectorizer works by converting a given set of strings into a frequency representation. For example, the text "The FDA approves a new drug that prevents migraines without side effects" would be converted into the table with the count of each word's frequency in it. This text example would have the 12 unique words and each would have a frequency of 1. Count vectors are most helpful in understanding the type of text by the frequency of the words in it. This method does not have the ability to identify the importance of words for analysis or identify the relationship between words in a text. The corpus was vectorized using the Bag of Words CountVectorizer method. The max_features and n_gram features were parameter tuned to find their optimal settings of a max feature set to 2,000 and an n gram range of 1-3.

After implementing multiple Naive Bayes classification models, the top 3 with the highest accuracy and precision scores were chosen to use in this experiment. The BernoulliNB, MultinomialNB, and ComplementNB models from the sci-kit learn package [20] are used. The data are imported from a CSV file with the 2,998 text records with their classification of fake or real. The fake and real labels were converted into numerical values, real being 1 and fake is assigned a 0 class. The data have been pre-processed and are split into training and testing sets with the scikit-learn package *train test split* [20]. The testing size was 33% of the data with a random selection.

The same models were reproduced to test the addition of the sentiment analysis feature. The data needed to be pre-processed again because the model was receiving text and numerical input. To accommodate our input types, a pipeline was created to use the FeatureUnion import. Following the work done in [21], we created a streamlined process to process the inputs accordingly. The pipeline, columnTransfer and FeatureUnion imports made it possible to preprocess numerical (sentiment scores) and textual data (health news) at the same time and combine together as input to the model. FeatureUnion works as if it makes a copy of the input data, and performs parallel transformers on the data. Each stream takes the same input, the numerical and textual data, and the output is concatenated. A diagram of the streams can be seen in Fig. 4. Stream A is the process the numerical input goes through, and stream B is the textual input. The pipeline makes it possible to transform each input differently, in parallel. The textual data are input in its preprocessed formatting, but it needs to go through the vectorization phase, and the character and token count processes. While the numerical data are put through an Imputer transformer to complete any missing values, this is done because the models will not run with missing data.

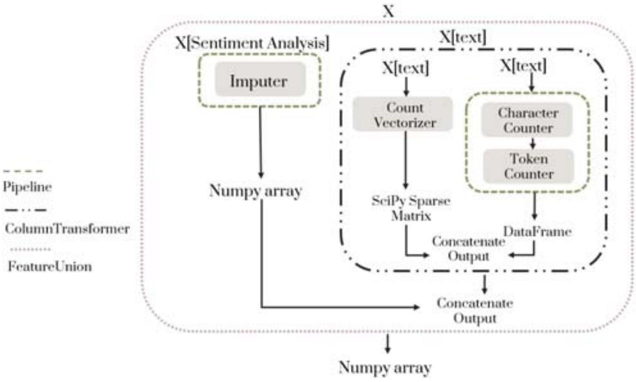

Fig. 4 FeatureUnion Used to Preprocess Input in Parallel

IV. Results

In this section we outline the results from the models, show the classification results such as accuracy, precision, recall and the F1 score of each tested model. We also explain the findings of the additional sentiment analysis feature and its impact on the precision scores. Finally, we end with topic analysis through an LDA model and explain the results of the sentiment analysis.

A. Model Comparison

The Bernoulli classification models with and without the sentiment analysis feature had the best precision score of 90.4% and 90% respectively. We are focusing on the precision and accuracy as the comparative metric to test if sentiment analysis improves the model. The precision score represents the amount of true positive classifications the model made. The nature of the content we are classifying is potentially dangerous information, this is why we put a bigger emphasis on precision score over accuracy. We would rather real information be classified as false over fake information being classified as real, that is where information can be dangerous. The confusion matrices of the models with and without the sentiment analysis feature are below. Fig. 5 shows that the fake class is being correctly classified most of the time. While the real data have a large amount of texts being classified incorrectly, this is seen as a safer alternative than mislabeling the fake data. However, we do want to increase the overall accuracy of the models in future work. Fig. 6 shows similar results with, majority of fake being classified correctly, while slightly over half of the real test data are classified as fake. Only 0.04% of the fake data are being incorrectly classified.

B. Sentiment Analysis Feature

Comparing the models after the addition of the sentiment analysis feature proved that the addition did not significantly improve precision or accuracy. Table IV documents all models metrics before and after sentiment analysis is added. All of the models saw increases in their precision score, the biggest increase being ComplementNB going from 82.1% to 84%. We also see that a pattern of the larger the precision score is without the sentiment analysis, the smaller the increase to

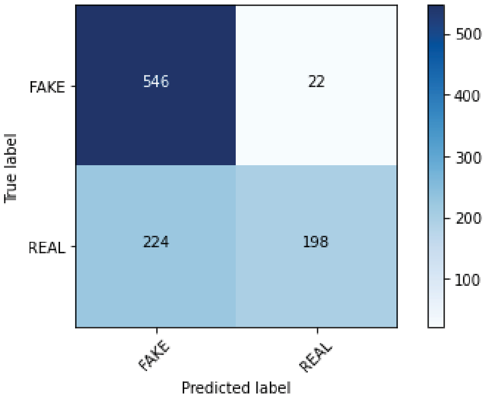

Fig. 5 Confusion Matrix of Bernoulli Naive Bayes Model *without* Sentiment Analysis Feature

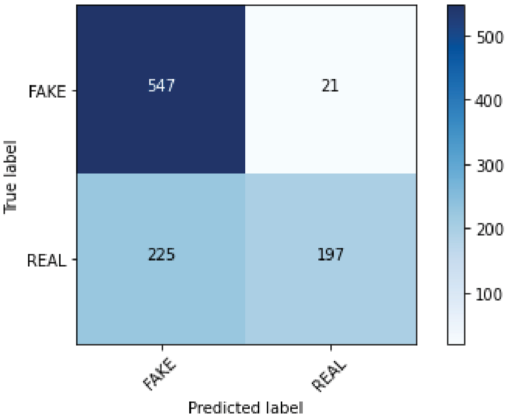

Fig. 6 Confusion Matrix of Bernoulli Naive Bayes Model *with* Sentiment Analysis Feature

precision is with the additional feature. Lastly, we saw that MultinomialNB had the greatest increase in accuracy, going from 73.3% to 74.1%.

TABLE IV
NAIVE BAYES MODELS PERFORMANCE METRICS *with* AND *without* SENTIMENT ANALYSIS

	Naive Bayes Model	Model Metrics			
		Precision	Accuracy	Recall	F1-score
Without Sentiment Analysis	Bernoulli	0.900	0.752	0.469	0.617
	Multinomial	0.838	0.733	0.464	0.598
	Complement	0.821	0.733	0.479	0.605
With Sentiment Analysis	Bernoulli	0.904	0.752	0.467	0.616
	Multinomial	0.855	0.741	0.474	0.610
	Complement	0.840	0.740	0.483	0.614

C. Topic Analysis

The goal of LDA is to find topics that a word belongs to within the dataset. We implemented LDA on the 3 sentiment polarities we generated with VADER. Similar preprocessing steps were taken before modeling including, removing stop words, removing non alpha numeric characters. Lastly, we removed any words that have a higher document frequency than 0.05%, as well as any words that have a lower than 10% frequency. These steps help remove noisy and infrequent words that might throw off the topic analysis. We selected 10 topics, they can be seen in Figs. 7-9.

LDA does not assign topics, it only finds the most related words that it finds and groups those into topics. As seen in Fig. 7, Topic 1 has a very high marginal topic distribution of over 10%. This metric can be thought of as the “importance” of each topic for the entire corpus, thus Topic 1 has a very high importance in relation to the entire positive dataset. The blue (lighter shade if viewing in black and white) bar graphs on the right represent the overall term frequency of the word, and the red (darker color) represents the estimated term frequency within the topic. The top 30 most relevant words are charted for each topic. We saw that the addition of the sentiment analysis feature did not show consistent improvement to the models. We investigated with LDA to find possible explanations as to why this feature did not add value. In all 3 positive, negative and neutral figures we see that the Topic 1, top words share similar terms such as, “cases”, “deaths”, “new” and “reported”. Since the contents of health information do not contain a lot of emotion in its content, sentiment analysis is not a significant indicator of a classification. There was no trend we identified within a sentiment distinguishing fake from real.

V. CONCLUSIONS

This work started with a data collection phase, and extensive preprocessing steps were taken to transform the data into the appropriate format for classification modeling with text. Next, multiple Naive Bayes models were implemented, first without the sentiment analysis feature, and then those same models were added to with the additional feature. We compared the precision and accuracy scores of the models before and after the additional feature to find the impact that sentiment analysis had on the models. The accuracy score of the BernoulliNB model stayed constant at 75.2%, while the precision score increased slightly from 90% to 90.4%. It can be concluded that on this dataset, sentiment analysis was not a valuable feature to this fake health news classification model.

VI. FUTURE WORK

This research can be expanded upon by completing a similar analysis, using a neural network as the classification model. Wang et al. [22] proposed an LSTM sentiment analysis model on short texts, this model could be tested with the addition of sentiment analysis on a fake health dataset to test the impact of sentiment analysis on the classification labeling. Another addition to this work would be repeating the same experiment with a larger dataset. Including more labeled news article titles found using web scraping might increase the model’s baseline classification accuracy. With more data to train on, the model might find more patterns in sentiment analysis and it could be a contributing feature. Lastly, a more in depth analysis using LDA could reveal more patterns within sentiment classified texts to further investigate the value of sentiment analysis on a health news classification model.

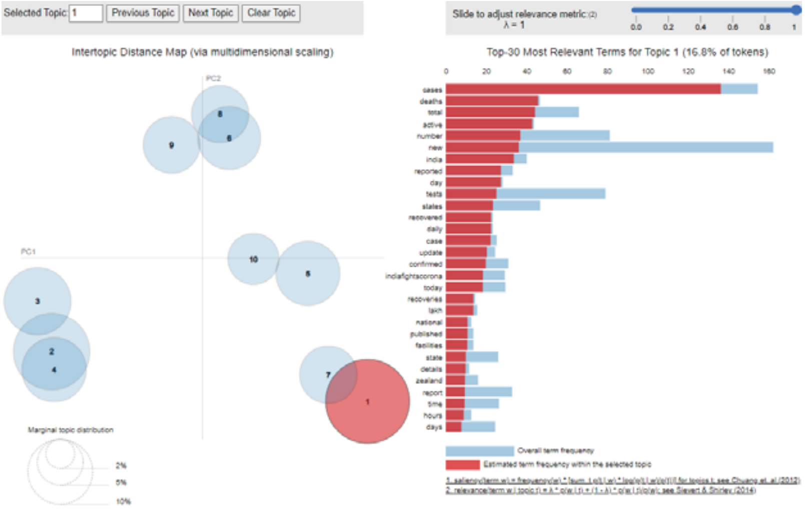
Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
8
9
10
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (16.8% of tokens)
0
20
40
60
80
100
120
140
160
cases
deaths
total
active
number
new
india
reported
day
tests
states
recovered
daily
case
update
confirmed
indiafightscorona
today
recoveries
lakh
national
published
facilities
state
details
zealand
report
time
hours
days
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 7 LDA Topic Analysis Visualization of Positive Dataset

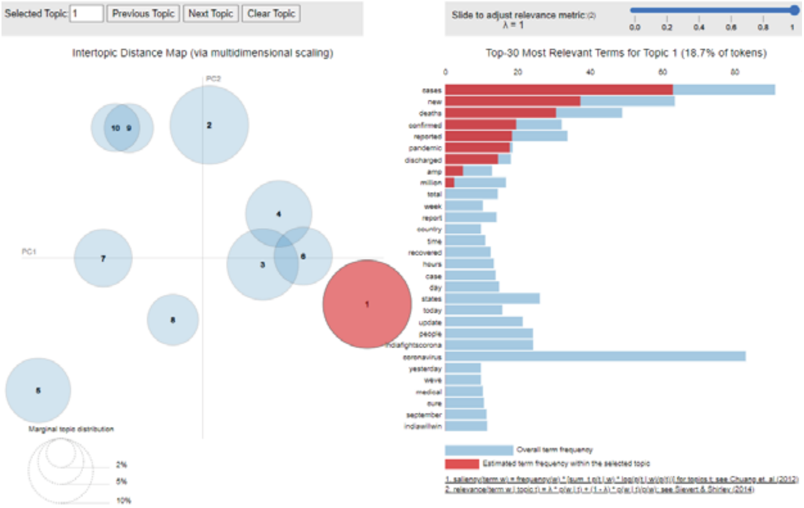
Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
10
9
2
4
7
3
6
1
8
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (18.7% of tokens)
0
20
40
60
80
cases
new
deaths
confirmed
reported
pandemic
discharged
amp
million
total
week
report
country
time
recovered
hours
case
day
states
today
update
people
indiafightscorona
coronavirus
yesterday
wave
medical
sure
september
indiawillwin
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 8 LDA Topic Analysis Visualization of Neutral Dataset

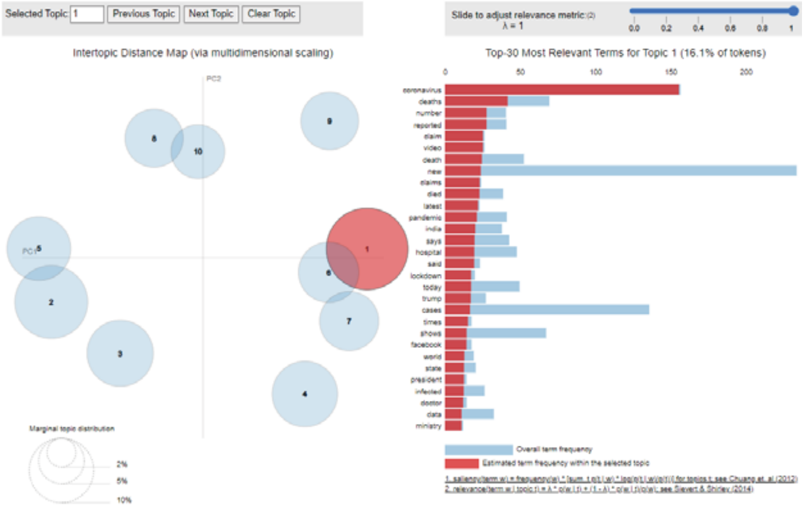
Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
8
10
9
5
1
6
2
7
3
4
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (16.1% of tokens)
0
50
100
150
200
coronavirus
deaths
number
reported
claim
video
death
new
claims
died
latest
pandemic
india
says
hospital
said
lockdown
today
trump
cases
times
shows
facebook
world
state
president
infected
doctor
data
ministry
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 9 LDA Topic Analysis Visualization of Negative dataset

References

[1] P. Coate, "Remote work before, during, and after the pandemic," Jan 2021.

[2] G. Juravle, A. Boudouraki, M. Terziyska, and C. Rezlescu, "Trust in artificial intelligence for medical diagnoses," in *PubMed*. National Library of Medicine, 2020. [Online]. Available: https://doi.org/10.1016/bs.pbr.2020.06.006

[3] T. Treharne and A. Papanikitas, "Defining and detecting fake news in health and medicine reporting," Aug 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7509617/

[4] K. Jiang and X. Lu, "Natural language processing and its applications in machine translation: A diachronic review," in *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, 2020, pp. 210–214.

[5] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[6] M. Razno, "Machine learning text classification model with nlp approach," *Computational Linguistics and Intelligent Systems*, vol. 2, pp. 71–73, 2019.

[7] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2017, pp. 900–903.

[8] S. Dhoju, M. Main Uddin Rony, M. Ashad Kabir, and N. Hassan, "Differences in health news from reliable and unreliable media," in *Companion Proceedings of The 2019 World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 981–987. [Online]. Available: https://doi.org/10.1145/3308560.3316741

[9] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20330466

[10] S. Elbagir and J. Yang, *Sentiment Analysis on Twitter with Python's Natural Language Toolkit and VADER Sentiment Analyzer*. Proceedings of the International MultiConference of Engineers and Computer Scientists 2019, 2019, pp. 63–80.

[11] E. Dai, Y. Sun, and S. Wang, "Ginger cannot cure cancer: Battling fake health news with a comprehensive data repository," *CoRR*, vol. abs/2002.00837, 2020. [Online]. Available: https://arxiv.org/abs/2002.00837

[12] E. Dai, "Fakehealth," 2020. [Online]. Available: https://github.com/EnyanDai/FakeHealth/pulls

[13] M. Porter, "Healthnewsreviews.org." [Online]. Available: https://tartarus.org/martin/PorterStemmer/

[14] H. S. Christopher D. Manning, Prabhakar Raghavan, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] S. Teja, "Stop words in nlp," 2020. [Online]. Available: https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47

[16] A. Beri. (2020) Sentimental analysis using vader. [Online]. Available: https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

[17] S. Panchal. (2020) Sentiment analysis with vader-label the unlabelled data. [Online]. Available: https://medium.com/analytics-vidhya/sentiment-analysis-with-vader-label-the-unlabeled-data-8dd785225166

[18] S. Saket. (2020) Count vectorizer vs tfidf vectorizer — natural language processing. [Online]. Available: https://www.linkedin.com/pulse/count-vectorizers-vs-tfidf-natural-language-processing-sheel-saket/

[19] S. Kannan, S. Saravanan, P. Chandirasekeran, and S. Rani Patra, "Detection of fake news related to covid-19 using natural language processing," in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 2021, pp. 1–6.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] Z. Luvsandorj. (2021) Featureunion, columntransformer and pipeline for preprocessing text data. [Online]. Available: https://towardsdatascience.com/featureunion-columntransformer-pipeline-for-preprocessing-text-data-9dcb233dbcb6

[22] T. W. L. Jenq Haur Wang, X. Luo, and L. Wang, "An lstm approach to short text sentiment classification with word embeddings," in *The Association for Computational Linguistics and Chinese Language Processing*, 2018.