Fake News Analysis and Detection

Sanket Joshi[1]

Abstract— **This project aims at analyzing some key characteristics and differences between real and fake news using the techniques learnt in the data mining class and harnessing them to develop classifiers that would work on any data corpus. We first clean the data, remove obvious indicative keywords, and then pass the data to various aggregate functions to compare class level characteristics across real and fake data. We develop a classifier purely based on these engineered features and test its accuracy.**

Next, we move on to using N-gram techniques to learn the representations of news articles and create a classifier using only the article titles. Eventually, we use the full article text to train our final set of models. We then pick the best performing model using an array of evaluation metrics. We discuss the trade-offs between these different levels of data inputs and resultant training time on one hand with the performance on the other hand.

Finally, we train a Recursive Neural Network with Long Short-Term Memory on the fake news data corpus to develop a fake news generator. We use the said best model to then test the accuracy of our generator. Needless to say, we only deal with a limited amount of data given the computational and time constraints. Also, there isn't a huge corpus of labelled data partially due to the objectivity of the person labelling the data itself.

I. INTRODUCTION

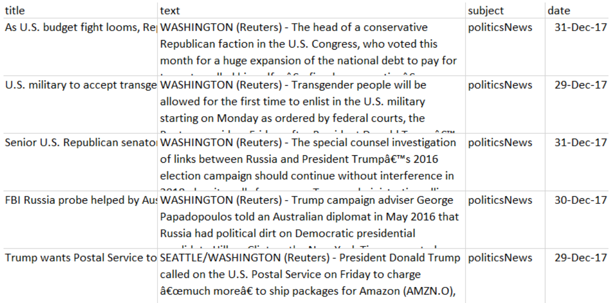
I would be remiss without quoting Neil Postman on this juxtaposition between George Orwell's "1984" and Aldous Huxley's "Brave New World".

"What Orwell feared were those who would ban books. What Huxley feared was that there would be no reason to ban a book, for there would be no one who wanted to read one. Orwell feared those who would deprive us of information. Huxley feared those who would give us so much that we would be reduced to passivity and egotism. Orwell feared that the truth would be concealed from us. Huxley feared the truth would be drowned in a sea of irrelevance."

The advent of the internet has seen an unprecedented rise in the number of alternative media sources. These often come with their own ulterior motives for political or commercial gain. Sensationalism and first-to-market strategies have not only affected alt-news, but sadly also been adopted by mainstream media sources to stay relevant. With the method of deceit not being as obvious as being detected by an Occam's razor, the giveaways are more subtle and often hidden in the selectivity of truths than blatant falsehoods. It has become impossible for fact checkers to keep up with the plethora of articles and blogs being published every day. Thankfully, machine learning and data mining techniques have stepped up to this task. Without further ado, let's move on to explore the relevant work.

The following section will describe the methodologies used to firstly analyze the key differences between real and fake news, and then build classification models that harness them. Although, it is important to note that whether a news is truly real or fake is not as objective as the data set makes it to be. There are different types and degrees of falsehoods. But for the scope of this project, we will assume that the data set has been labelled accurately. Let us first look at the data set.

II. The Data Set



title	text	subject	date
As U.S. budget fight looms, Re	WASHINGTON (Reuters) - The head of a conservative Republican faction in the U.S. Congress, who voted this month for a huge expansion of the national debt to pay for	politicsNews	31-Dec-17
U.S. military to accept transge	WASHINGTON (Reuters) - Transgender people will be allowed for the first time to enlist in the U.S. military starting on Monday as ordered by federal courts, the	politicsNews	29-Dec-17
Senior U.S. Republican senato	WASHINGTON (Reuters) - The special counsel investigation of links between Russia and President Trumpâ€™s 2016 election campaign should continue without interference in	politicsNews	31-Dec-17
FBI Russia probe helped by Au:	WASHINGTON (Reuters) - Trump campaign adviser George Papadopoulos told an Australian diplomat in May 2016 that Russia had political dirt on Democratic presidential	politicsNews	30-Dec-17
Trump wants Postal Service to	SEATTLE/WASHINGTON (Reuters) - President Donald Trump called on the U.S. Postal Service on Friday to charge â€œmuch moreâ€ to ship packages for Amazon (AMZN.O),	politicsNews	29-Dec-17

Fig. 1. Sample raw file

[1] Sanket Joshi is a Computer Science Graduate student at Dartmouth College, Hanover, NH 03755, USA `sanket.s.joshi.gr@dartmouth.edu`

The data set used in this project is the Kaggle's fake and real news data set. The data spans from March 2015 to February 2018 and contains American news articles from mainstream news as well as independent blogs. It consists of two CSV files, with 21417 real articles and 23450 fake articles, respectively. Each article contains a title, the text, the subject, and the published date.

A. *Cleaning*

The articles between the two categories differ in certain indicative marks for the instance, the real articles contain Reuters sources like "WASHINGTON - REUTERS", "NEW YORK - REUTERS", etc. The fake articles on the other hand have text appended to media captions like "[IMAGE]", "(VIDEO)", "[Tweet]", etc. Our first task would be to remove these simple indicators that would otherwise produce amazing results even with the simplest rule-based algorithms.

An efficient way to find such marks is to check the most frequent terms and remove any that are too specific to the data set and irrelevant to the actual news. This not only helps in generalizing the model for new data sets, but also makes our models more robust to over-fitting. We also notice the presence of certain non-ASCII characters that can be easily removed by standardizing our encoding throughout the data.

B. *Preprocessing*

Several methods of preprocessing were considered and weighed while studying this data set. After looking through various tokenization methods, we went with the NLTK's regular expression tokenizer due to its speed. This decision was made after experimenting with a subset of data and extrapolating the processing time to the larger data set. The marginal performance increase considering the drastic increase in processing time was not worth the effort. With the overall size of data, it made more sense to invest time in training and model selection rather than preprocessing. Lower casing was applied to the entire data.

While considering our options between lemmatization and stemming, once again, we extrapolated that the former would take much longer time, and the later, despite of its crudeness, still takes more than an hour to run on the entire data set, while marginally increasing performance. As mentioned, time spent in model selection to test out different parameter combinations is much more fruitful than that in preprocessing.

III. Analysing Fake News

Before we attempt to build a classifier, let's first try to analyze fake news and look at certain key characteristics that can help us build engineered features rather than having to parse through the entire article for building our models.

A. *Temporal Distribution*

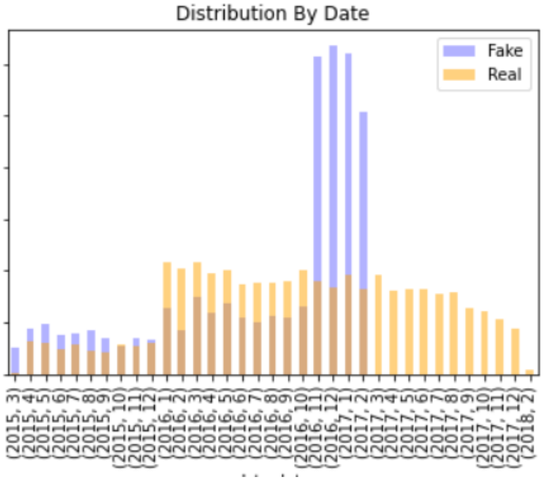


Fig. 2. Temporal Distribution

Looking at the distribution of data over time, we plotted this histogram and made an immediate observation regarding the anomalous nature of fake news. It seems like there is a sharp peak during late 2016 which was the period of US elections. This is also revealed by the fact that most of the fake news was political in nature looking at the subject column. We made it a point to omit data from this column as well since it was too indicative of the article class.

B. *Lexicographical Distribution in Titles*

This method of analysis is one of the most revealing methods in terms of understanding the intent, the efforts, the formalism, and biases in news articles.

- **Word Cloud:**

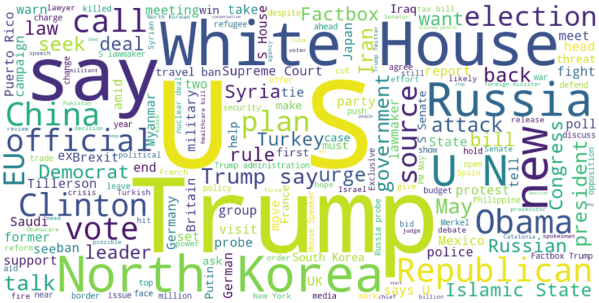


Fig. 3. Most common article titles - Real News

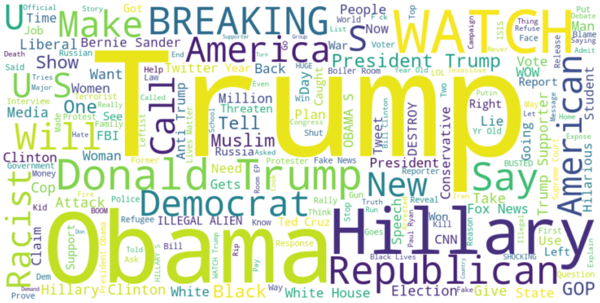


Fig. 4. Most common article titles - Fake News

Generating a word cloud is always a good way to give a cursory look into this as well as develop further intuitive ideas as to what engineered features we can extract (Fig. 3 & Fig. 4). We see that fake news revolves more around political factions and figures related to American politics whereas real news is a has a better representation of global politics. We also see the absence of certain political candidates in real news, depicting the biased nature and hidden agenda behind fake news.

- **Word Frequency:**
  Next, we plot the word frequency of the top 50 most frequent words in the titles of both news (Fig. 5). Although we expect an exponential distribution, the peak for fake news is significantly sharper, and falls pretty rapidly after the first 3-5 words. We still have a big chunk within the top 20 words where fake news still has a lead. But eventually, these frequencies between the two classes seem to even out.

- **Lexical Diversity:**
  Judging from the previous distribution patterns, we would intuitively say that fake news has lesser lexical diversity than real news due to the narrower topics to handle. But on the contrary, we see the following results.

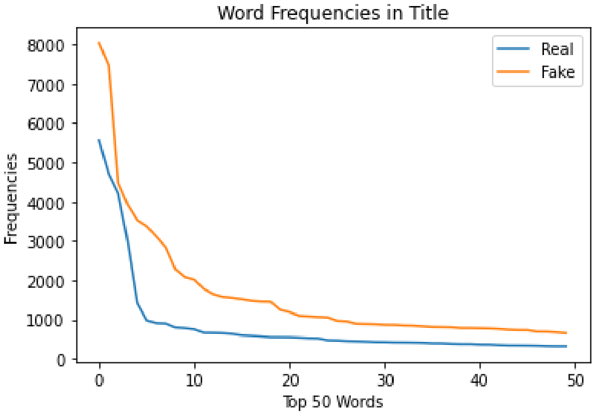


Fig. 5. Word frequency in titles

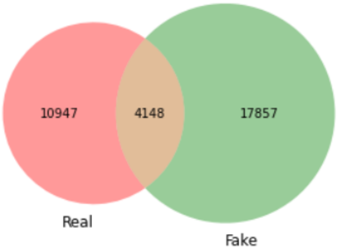


Fig. 6. Vocabulary in Titles

  Fake news contains roughly 63% more diverse words than real vocabulary. Despite of the narrow scope of topics, one of the reasons for this could be the usage of more informal language as well as the presence of typing errors and profanities. The size of intersecting words being so low is also a surprise.

- **Word Exclusivity:**
  The following are a few of the most frequent words exclusive to real article titles: *termination, georgetown, frontrunners, waving, grills, g40, prohibition, nuances, apprentice, adha, institutions, clauses, profiling, cano, unmatched*
  On the other hand, the following are a few of the most frequent words exclusive to fake article titles: *airfare, defilement, cashing, kahn, jewelers, retaliate, naturalization,*

nutjob, smuggled, aliens, diagnoses, hatemonger

Although the divide is not clear, it could be indicative of the fact the real differences would be apparent in the article body rather than the titles. It could be the case that certain words are thematically unsuitable for titles in terms of serving as clickbaits.

C. Lexicographical Distribution in Article Body

Now, we do similar analysis on the words in article body, and see if they differ in the same way as in the titles, or have a slight nuance.

- **Word Cloud:**

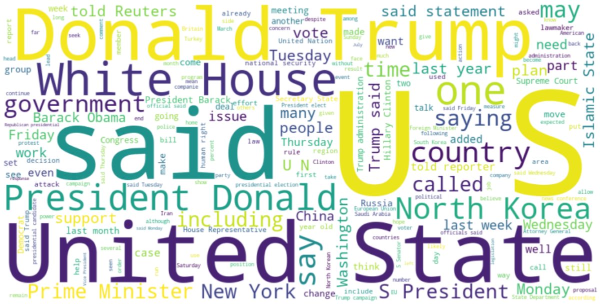


Fig. 7. Most common article body words - Real News

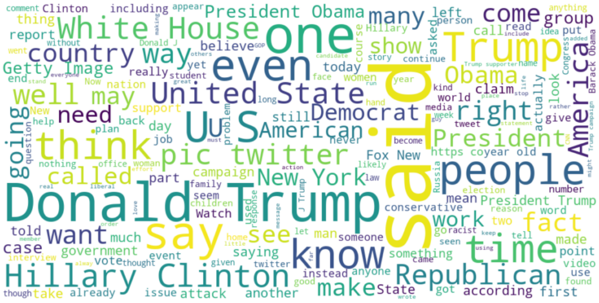


Fig. 8. Most common article body words - Fake News

The results here are again surprising (Fig. 7 & Fig. 8). One would think that fake news would have a sharper exponential distribution than real news in terms of the article body as well. But this is not at all the case. This might be partly due to the redundancy of fake news in reinforcing certain points versus the succinctness of real news due to more formal sources. However, we see that the top few words share a lot of commonalities between the two classes.

- **Word Frequency:**

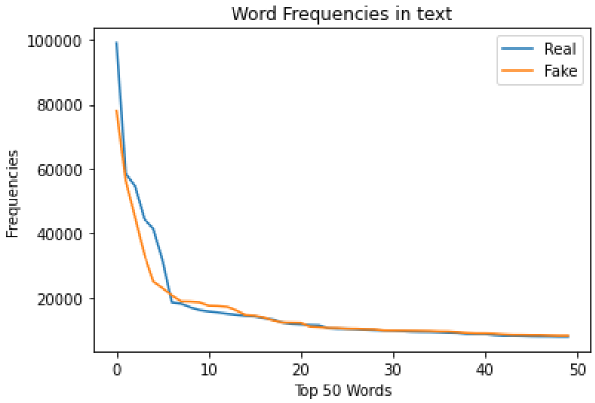


Fig. 9. Word frequency in article body

Next, we plot the word frequency of the top 50 most frequent words in the article bodies of both news (Fig. 9). The results here are indicative of our observations from the word cloud, with real news having a sharper peak. Although it is surprisingly similar after the top 10 words.

- **Lexical Diversity:**

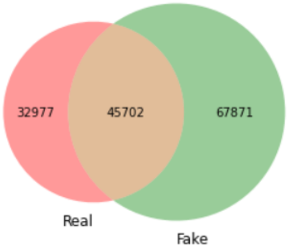


Fig. 10. Vocabulary in Article Body

The differences here are even more visible. Fake news body contains roughly 106% more diverse words than real news. This fact is maintained and let us reiterate that fake news seems to use more informal language.

- **Word Exclusivity:**
In terms of the article body, this factor is not much different, apart from the fact that the amount of profanities and jargon increases dramatically in fake news, whereas

the amount of technical terms increases in real news.

D. *Sentiment Analysis*

We now move on to analyzing the distribution of sentiments in the article titles across real and fake news. Note that we exclude the sentiments whose score show up as zero, but also, that most of the titles (around 86%) score a zero on the positive, negative, and compound side. The histograms represent the distribution of non-zero scores and are more of a comparison between the classes than of between sentiments. On a broader scale, the sentiments seem to match across classes, but the area of curve in greater for fake news when put together across sentiments. This imbalance is explained by the exclusion of sentiments with the score of zero, which is more prevalent in real news.

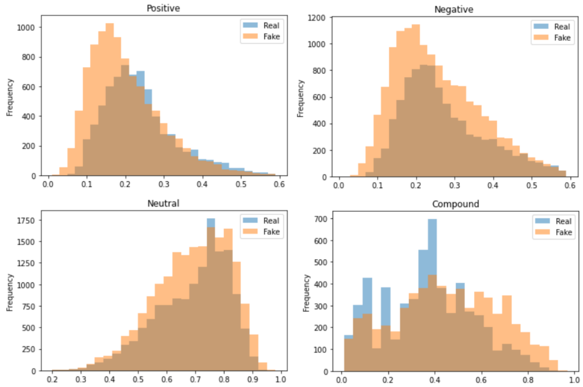


Fig. 11. Sentiments in Titles

- **Polar Sentiments:**
  The polar sentiments (positive & negative) seem to have a peak at lower magnitudes representing that if titles do have sentiments, they are not extreme. In general, fake news has a greater area under the curve representing that more fake titles are sensitive. However, their overall peak is at a lower magnitude than that of real news, indicating that if they do have a sentiment, it is more subtle. This might be because of the fact that lies or subtle deceits do not necessarily require polar language, and can be plain statements.

- **Non-Polar Sentiments:**
  The neutral sentiment on the other hand is the most prevalent sentiment overall and also has a peak at a higher value. However, contrary to belief, more fake titles are neutral than real titles looking at the area under the curve. There aren't clear patterns when it comes to compound sentiments, but fake news has an overall higher magnitude.

IV. Extracting Features

Before developing a full-fledged classifier that works on N-gram analysis, let us first develop one that is computationally faster but reaches comparable accuracy levels. We extract some numerically engineered features and feed them to an array of models with different parameters. Let us take a look at these features.

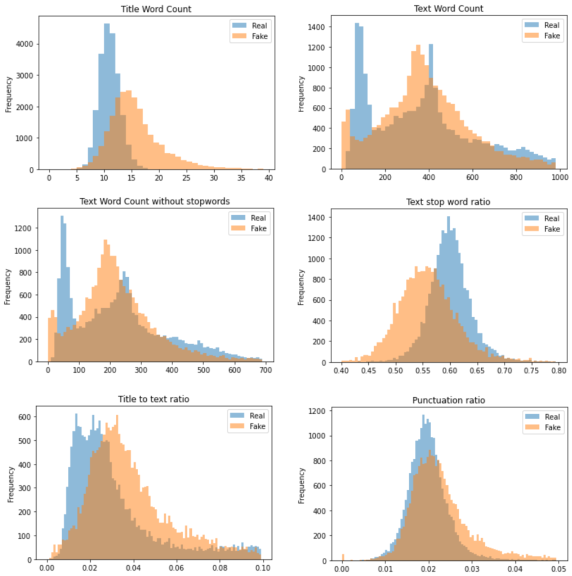


Fig. 12. More Engineered Features

A. *Title Word Count*

The title word count has a class-wise normal distribution and is a really good classifier, given that most of the fake news title lengths are longer than the real news mean. The reason could very well be the attempt of fake news to convey most of the information in the title itself to serve as the perfect click-bait.

B. Text Word Count

Fake news article length follows a near-normal distribution, but real news has two peaks indicating that the content length is either very short, or approximately as long as fake articles. The former could be due to news that genuinely does not have much information to convey and the later could be the norm.

C. Text Word Count without Stopwords

The length of article body after removing all the stop words would indicate the level of redundancy. We see that real news articles that are shorter have approximately the same distribution of length without stop words, but the lengthier articles shrink considerably. This does not happen to fake news, where the distribution is similar in shape.

D. Text minus stop words to Text Ratio

The proportion of text minus stop words to raw text is indicative of the succinctness of the article content. The results here are as expected and rather perfectly separate out the two classes with two bell curves that are slightly overlapping. Real news is more succinct and fake news is more redundant

E. Title to Text Ratio

Once again we see that fake news not only has longer titles by itself, but also has slightly longer titles relative to the article body, once again conveying the intent to express most of the information in the title.

F. Punctuation to Text Ratio

This is not much of a useful statistic since the curves overlap. Although, real news has a higher magnitude overall. It would definitely be interesting how this would fare if we broke down the different punctuation characters.

Indeed, we might have struck gold here?! (A practical joke). Exclamation marks and question marks are almost exclusively found in fake news. This would serve as an Occam's razor.

G. The final dataframe

Given all these engineered features, we now show the enhanced data frame. (Fig. 14)

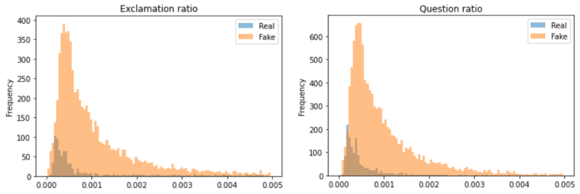
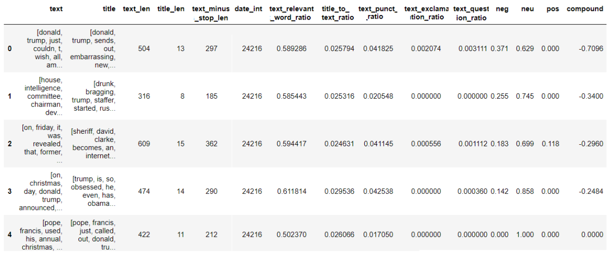


Fig. 13. Exclamations and Question Marks



	text	title	text_len	title_len	text_minus_stop_len	date_int	text_relevant_word_ratio	title_to_text_ratio	text_punct_ratio	text_exclamation_ratio	text_question_ratio	neg	neu	pos	compound
0	[donald, trump, just, couldn, t, wish, all, am...	[donald, trump, sends, out, embarrassing, new,...	504	13	297	24216	0.589286	0.025794	0.041825	0.002074	0.003111	0.371	0.629	0.000	-0.7096
1	[house, intelligence, committee, chairman, dev...	[drunk, bragging, trump, staffer, started, rus...	316	8	185	24216	0.585443	0.025316	0.020548	0.000000	0.000000	0.255	0.745	0.000	-0.3400
2	[on, friday, it, was, revealed, that, former, ...	[sheriff, david, clarke, becomes, an, internet...	609	15	362	24216	0.594417	0.024631	0.041145	0.000556	0.001112	0.183	0.699	0.118	-0.2960
3	[on, christmas, day, donald, trump, announced,...	[trump, is, so, obsessed, he, even, has, obama...	474	14	290	24216	0.611814	0.029536	0.042538	0.000000	0.000360	0.142	0.858	0.000	-0.2484
4	[pope, francis, used, his, annual, christmas, ...	[pope, francis, just, called, out, donald, tru...	422	11	212	24216	0.502370	0.026066	0.017050	0.000000	0.000000	0.000	1.000	0.000	0.0000

Fig. 14. Data Frame with Engineered Features

V. Classification with Engineered Features

It is finally time to test out different models on the features that we've built. Let us look at the results.

A. Results

We take different variants of each model and rank order them based on the test scores. We display the top variant of each model type and see that `RandomForestClassifier` gives us a test accuracy of 94.597%.

Model Type	5 Fold Cross Validation			Testing
	Acc.(%)	F1(%)	Wt. F1(%)	Wt. F1(%)
Random Forest	94.328	94.513	94.328	94.597
Decision Tree	91.372	91.695	91.369	92.130
K-Neighbors	85.519	85.763	85.523	86.394
Logistic Reg.	81.827	81.750	81.824	81.419

TABLE I
Results on Engineered Features

B. Confusion Matrix

We see the confusion matrix in Fig. 15., where we notice that false Negatives have a slightly higher contribution to the error than false positives. But this is preferable since we would rather under-detect fake news than mark some real news as fake.

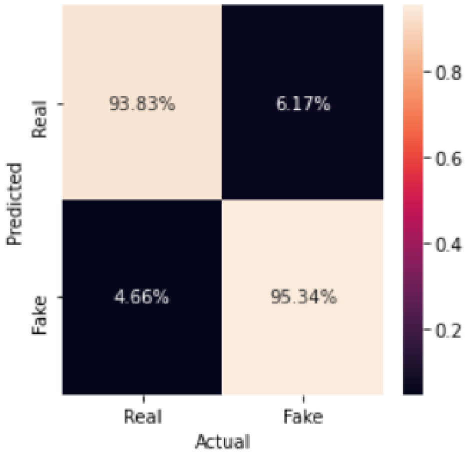


Fig. 15. Confusion Matrix

C. *Feature Importance via Elimination*

Let us do some analysis using various approaches to find out which features contribute most to the model accuracy. Let us find out the dip in accuracy by excluding each feature individually to find their importance.

We see that this exercise is not that helpful since we do not see a drop of more than 2% (Fig. 16). In fact, the last 5 features when dropped increase the accuracy. Although, this could partly be due to the randomness of results across tests, so let us not drop them altogether.

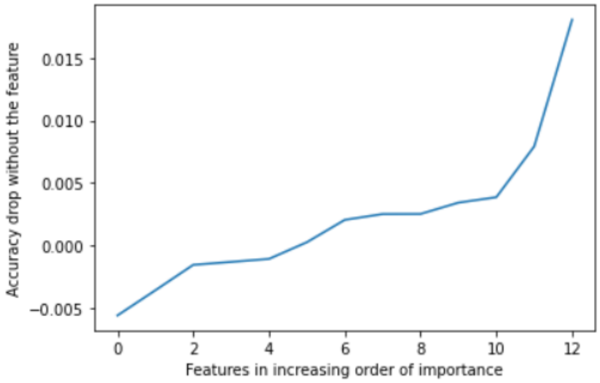


Fig. 16. Accuracy drop with individual feature exclusion

D. *Feature Importance via Selection*

Since elimination was not that revealing, let us find out the dip in accuracy by including only each individual feature to find their importance (Fig. 17, Table II). We see that this exercise is much more helpful and that features like title length, date, and relevant word ratio are the most important features. In fact, using only the title length gives an accuracy of 78.85% which is pretty impressive!

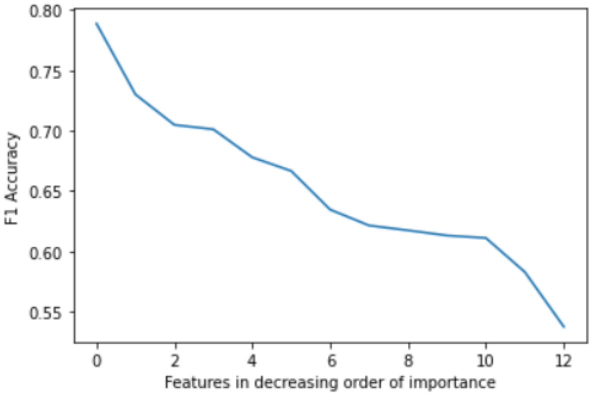


Fig. 17. Accuracy with singleton features

Singleton Feature	F1 Score(%)
title_len	78.85
date_int	72.98
title_to_text_ratio	70.46
text_relevant_word_ratio	70.09
text_question_ratio	67.77
compound	66.63
neu	63.43
text_minus_stop_len	62.11
neg	61.71
text_punct_ratio	61.28
text_len	61.08
text_exclamation_ratio	58.26
pos	53.72

TABLE II
ACCURACY WITH SINGLETON FEATURES

E. *Cumulative Performance*

Now that we have a reliable ordering of the top few important features, let us see if we can afford to drop some features in anticipation for very large data sets. Let us include the first N most important features obtained via selection to see when the performance plateaus to determine what features we can ignore completely. We vary N from 1 to total number of features, i.e. 13.

We see that this exercise is very revealing that including the first 4 features gives us comparable accuracy (94%) to including all 13 (Fig. 18). And as noticed in the elimination section, we see the

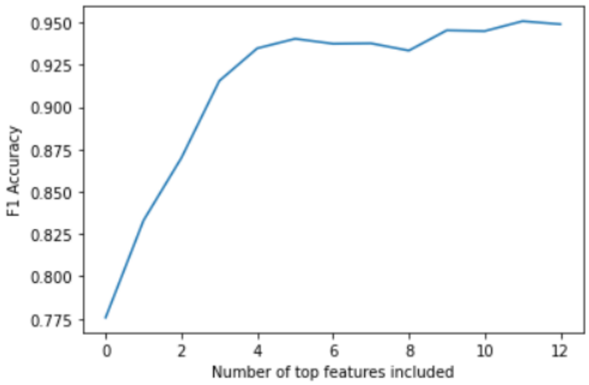


Fig. 18. Accuracy with top N features

instability of accuracy after that point, as well as diminishing returns. This can be harnessed for larger data size.

VI. Classification with N-Gram Analysis

Now that we have created a cheaper classifier using only numerical features and achieved up to 94.6% accuracy, let us see how much we can improve upon this by using a combination of models with different N-Gram size ranges, and count or TFIDF vectorizers. But this time, we only use the titles since the vocabulary and vector size if trained on full text would very time consuming. Once we have the top model, we then use that model to create a classifier with the entire article body, assuming that it is also optimized for that. Let us take a look at the results.

A. Results with only the titles

Let us look at the model variants along with the terminology used (Table III). We use 5 models - Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN3 & KNN5). For each model, we either use a Count Vectorizer (TFIDF=0) or a TDIDF Vectorizer (TFIDF=1). Further, we either use only unigrams (NGrams=1), or a unigram and a bigram (NGrams=2) or all 3 of Unigrams, Bigrams and Trigrams (NGrams=3).

For each of these combinations, we show the accuracy (Acc.), the F1 Score (F1) and the Weighted F1 Score (WF1) for 5-fold Cross Validation. This is then followed by the Weighted F1 Score on the test set (WF1). Finally, we sort in descending order of the test score (Last column).

We see that the best performing model is Linear Regression without TFIDF that uses all N-Grams, with a test score weighted F1 of 96.03%. Overall, Linear Regression and SVM perform really well and KNN's perform poorly.

Model	Parameters		5-fold CV (%)			Test(%)
Type	TFIDF	NGrams	Acc.	F1	WF1	WF1
LR	0	2	95.80	95.90	95.80	96.03
SVM	0	2	95.15	95.27	95.15	95.93
SVM	1	2	94.90	95.05	94.90	95.87
SVM	1	3	94.86	95.02	94.86	95.73
LR	0	3	95.76	95.86	95.76	95.58
SVM	0	3	95.19	95.30	95.19	95.52
LR	0	1	95.27	95.38	95.27	95.48
SVM	0	1	95.07	95.17	95.07	94.90
LR	1	1	93.33	93.52	93.33	94.09
LR	1	2	93.32	93.54	93.32	93.98
LR	1	3	92.96	93.21	92.95	93.86
SVM	1	1	94.04	94.20	94.04	93.85
RF	0	1	91.69	91.87	91.69	92.99
RF	0	3	91.94	92.05	91.94	92.92
RF	0	2	92.11	92.27	92.12	92.91
RF	1	1	91.36	91.45	91.37	92.24
RF	1	2	91.86	91.97	91.86	90.53
KNN$_5$	1	3	88.44	88.56	88.44	89.14
RF	1	3	92.06	92.18	92.06	89.11
KNN$_5$	1	2	88.39	88.53	88.39	88.80
KNN$_5$	1	1	88.08	88.03	88.08	88.34
KNN$_3$	1	2	87.53	87.67	87.54	88.27
KNN$_3$	1	1	87.08	87.02	87.08	88.10
KNN$_3$	1	3	87.89	88.09	87.89	87.46
KNN$_5$	0	2	72.92	69.87	72.52	73.08
KNN$_5$	0	1	71.94	71.86	71.89	72.50
KNN$_3$	0	1	69.79	70.37	69.76	71.10
KNN$_5$	0	3	70.06	71.22	70.03	70.34
KNN$_3$	0	2	70.01	67.42	69.68	69.26
KNN$_3$	0	3	67.01	67.51	66.61	66.11

TABLE III

N-Gram Classification on Titles

B. Results with the entire article body

Model	Parameters		5-fold CV (%)			Test(%)
Type	TFIDF	NGrams	Acc.	F1	WF1	WF1
LR	0	3	98.88	98.91	98.88	99.18

TABLE IV

N-Gram Classification on Text

Now that we have the top performing model on titles, we use the model parameters to train the model on the entire article body of the data set and see the final results (Table IV). Note that this process creates a vectorizer with hundreds of thousands of dimensions and hence is very slow to train. This being the case, we only train using the parameters obtained in the top model from training on the titles.

Albeit, we achieve a test weighted F1 score of 99.18% which is still pretty impressive. Needless to say, a model would ideally be trained using all combinations of parameters but it is not really worth the additional increase in accuracy if we are limited by time and hardware constraints.

VII. Fake News Generation using RNN/LSTM

Now that we have a well performing fake news classifier, let us train an existing LSTM on the corpus of fake news to generate a text paragraph. Note that this is not an actual exercise but just a thought experiment. For more details, check the Jupyter Notebook.

VIII. CONCLUSIONS

We have presented in this project some revealing characteristics about fake news. We have seen some characteristic repetitions, clustered temporal distribution, highly exponential word frequency, diversity in vocabulary, informal tone, and, heavy use of interjections, interrogations, and profanities. We have harnessed these differences to explore various combinations of models and parameters to pick the best classifier.

We have used n-gram analysis to improve upon the said classifiers by first using build a classifier on article titles with a 96% accuracy. Next, we have used n-gram analysis on the article body to build a classifier with 99+% accuracy, and found out that Linear Regression with count vectorizer, and unigrams, bigrams, and trigrams works the best.

IX. FUTURE WORK

Looking at the assumptions we made during the start of this project regarding the data set, we can definitely point out that the problem of fake news detection is not a simple binary classification problem in real life. We can look at the general vocabulary and formatting signatures of an article and determine its validity, but its true validity can only be determined using fact checking methods. What we are doing right now is that we are determining this validity by the stylized approach of the author. This model can actually be used by an organization to ensure that the real news that they are trying to publish does not look fake, or for adversaries, the fake news that they are trying to publish at least passes the low level litmus test.

The future work on this area can involve extracting the vectors from a given statement that could be a placeholder for its claims, and then looking them up on a range of vectorized articles that are from credible sources. We would then check for similarity scores with these articles and see if the statement is not saying something that is out of the ordinary. However, such article aggregators are generally commercial products that are used by large organizations and would not be in the scope of this project. Either ways, there is still a great deal of promising research pending in this field.

X. ACKNOWLEDGMENTS

Thank you to Prof. Soroush Vosoughi for a great practical introduction to Data Mining and for imparting an intuition to various tips and tricks in the field.

References

[1] Julio C. S. Reis et. al., Supervised Learning for Fake News Detection, IEEE Intelligent Systems (Volume: 34, Issue: 2, March-April 2019), DOI: 10.1109/MIS.2019.2899143

[2] Xinyi Zhou et. al., Fake News: Fundamental Theories, Detection Strategies and Challenges, WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, https://doi.org/10.1145/3289600.3291382

[3] Ray Oshikawa et. al., A Survey on Natural Language Processing for Fake News Detection, Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020) pp. 6086-6093, arXiv:1811.00770

[4] Shivam B. Parikh et. al., Media-Rich Fake News Detection: A Survey, 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), DOI: 10.1109/MIPR.2018.00093

[5] Mykhailo Granik et. al., Fake news detection using naive Bayes classifier, 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), DOI: 10.1109/UKRCON.2017.8100379

[6] Data Source: Kaggle - Fake and real news dataset (https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset)