Machine Learning Professionals HATE Them!!! Genius Princeton Students Discover **Miracle** Method to Identify Fake News

The Real Deal with Fake News

Ilene E
Princeton University
ilenee@

Ze-Xin Koh
Princeton University
zkoh@

Christine Kwon
Princeton University
cmkwon@

Eileen Wang
Princeton University
ew23@

Abstract

Anyone can recognize the sensational fake headline, like this paper's title, riddled with unnecessary capitalization, exaggerated language and overwhelming punctuation. However, not all fake news is so easy to identify. In this assignment, we intend to address the challenge of classifying news articles into "real" or "fake" news utilizing a number of different methods for classification. We examined the difference between a bag of ngrams representation and features extracted with Parts-of-Speech tagging. We will be evaluating these methods on a dataset of political news headlines from kaggle.com, which contains 10,223 statements from various political headlines and labels them as either "real" or "fake". We found that since many of the phrases used in real and fake news are similar, the consideration of syntactic structure using Parts-of-Speech tagging is promising in the improvement of classifier performance.

1 Introduction

With the viral power of social media today, fake news spreads faster than ever. This phenomenon poses several potential dangers including "the misallocation of resources during terror attacks and natural disasters, the misalignment of business investments," [3] and, as seen across multiple countries in recent years, misinformation in political elections. Thus, we are interested in understanding which phrases or linguistic structures best distinguish between real and fake news headlines. This would inform the choice of the most helpful classification methods for distinguishing between reliable and unreliable news sources. We would like to understand the properties of a classifier that are useful in performing this task in order to consider developing new classifiers specifically for this purpose.

The challenges of this task lie in the complexities of linguistic patterns. Even a human user requires prior knowledge and the assessment of context to successfully classify a news headline as real or fake. Generally, the detection of fake news is a much more involved task than other text analysis tasks such as sentiment analysis, as it is usually insufficient to detect for "positive" or "negative" words. Many of the individual words or phrases found in real news are found in fake news as well. The reason we, as human users are capable of distinguishing between the two is because we are able to draw links between the different words in a statement to determine its plausibility. Given the large quantity of different links that could be made between words in any given sentence, it will be undeniably challenging to train a model to detect the significant semantic difference between fake and real news.

2 Related Work

A paper by Oshikawa and Wang gave an overview of current common practices and their limitations with regards to fake news detection and similar tasks, like fact checking and rumor detection [6]. The use of Parts of Speech (POS) tagging to break sentences into meaningful linguistic chunks is definitely more effective than the use of n-grams in the detection of patterns among fake news [10]. However, this practice comes at a huge performance trade-off, making it mostly intractable on the large datasets required for a model to gain sufficient context to make meaningful predictions.

Many fake news detection algorithms that have been developed take into account factors besides the actual text, such as the website it's from, and statistics such as the like, comment, or repost counts [9]. Some even check how long the website has been around for, as most fake news websites are young [9]. These supplementary sources of information are useful as some sites have a history of producing faulty news articles, and the text of many fake news articles do not provide strong cues as to their legibility.

3 Methods

3.1 Data Description

The dataset we used was obtained from Kaggle [1]. It consists of 10223 news statements scraped from PolitiFact.com [2]. PolitiFact.com is a website that fact-checks political headlines from different news sources, and assigns each statement a truth label. 56% of the statements in the dataset are real and the rest are false. The classes are largely balanced.

3.2 Data Processing

We used a 80/20 ratio to split the dataset into a training and a testing set, of sizes 8192 and 2048 respectively. Each sentence was tokenized into words, and each word was stemmed and lemmatized, i.e. suffixes were removed so that "supports" and "supporting" would be considered the same token by models later on. We then used a bag-of-words representation to represent each statement. The vocabulary produced consisted of 12522 unigrams, bigrams and trigrams, that occurred more than once in the dataset.

3.3 Latent Dirichlet Allocation (LDA)

In our study, we utilized Latent Dirichlet Allocation in order to examine common topics of real and fake news, which could be used to inform choices in hyperparameters of our classification models. We wanted to see if both types of news covered a similar range of topics, or if they had tendencies to focus on different things. First we created a bag of words utilizing the CountVectorizer function from sklearn with 1000 feautures and removing stop words. Then we ran LDA with the default parameters and printed out the top 5 derived topics, each with 10 top words.

3.4 Classification

We used a number of different classifiers from the SciKit Learn libraries [4]. All hyperparameters are the default unless specified:

1. Bernoulli Naive Bayes (NB)
2. Support Vector Machine (SVM)
3. Random Forests (RF)
4. Multi-Layer Perceptron (MLP)
5. Decision Trees (DT)

We chose SVM and RF for our classification task because they are known to perform well on similar text classification tasks [7] ([8]. Since RF is essentially an ensemble classifier that takes the average of the results of many Decision Trees, we decided to include DT as a baseline method to

compare the significance of results obtained with RF. We also chose MLP as we wanted to see how a neural network based model would compare to the other statistical models.

In addition, we used a neural network model from the SpaCy library, which processes words using Parts-Of-Speech (POS) tagging [5]. In Parts-Of-Speech (POS) tagging, each word in a sentence are tagged as nouns, verbs, adjectives, and so on [11]. This allows noun phrases, "subject-verb" and "verb-object" phrases to be grouped into meaningful linguistic blocks. As shown in Figure 1, dependency labels describe connections between words in a sentence, which allows the classifier to determine which similarities between an observed training sample and a new test sample are worth pursuing. These dependency labels also narrow down the number of combinations that the model would have to consider during a prediction, which greatly reduces computation time.

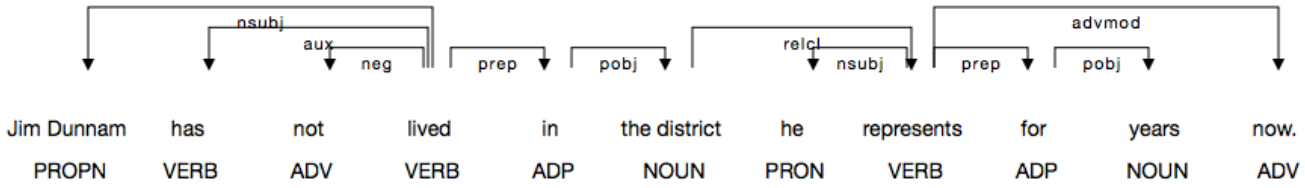

Figure 1: PoS tags and dependency labels that describe relationships between words in one of the headlines in our dataset.

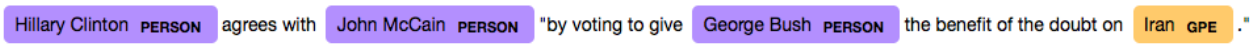

Figure 2: The recognition of named entities such as people and locations, within a sentence.

3.5 Evaluation Metrics

We use evaluation statistics that utilize the numbers of true/false negatives and positives. Specifically, we focus on precision, recall and specificity. By convention, denoting the number of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN), we define:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

We include specificity as well as recall because we believe false positives are important, since they would lead to users believing information that is false. Paying attention to all three of these evaluation metrics prevents our analysis from being affected by the slightly imbalanced classes.

4 Results and Analysis

4.1 Words most highly correlated with Fake and Real News

Fake	Real
Obama	Incarcerate
Scott Walker	Time
President	Decade
Hillary Clinton	Job Growth
Murphy	Year

The table shows phrases that are highly correlated with fake news and with real news.

This collection of words correlated with fake news suggests that since the names of important political figures provide eye-catching headlines, they are commonly used in fake news, or in smear campaigns by the opposition party. On the other hand, words correlated with real news appear to be more mathematical, factual phrases, that are neither extreme nor clickbait in nature. These results suggest that the assumption that words in a headline are independent, which is implicitly used by a bag-of-ngrams representation, is justifiable.

4.2 Exploratory Data Analysis: Latent Dirichlet Allocation

Here are our results from performing LDA with 5 topics and 10 features on real, fake, and combined news.

The top 5 topics for fake news were as follows:

topic 0	obama says president barack 000 percent year years obamacare security
topic 1	people health percent says pay care voted tax taxes like
topic 2	says illegal dollars tax million did city america county bush
topic 3	says state wisconsin care scott health jobs 000 clinton billion
topic 4	says states texas united school dont rick know public state

The top 5 topics for real news were as follows:

topic 0	000 jobs state new year million public office oil clinton
topic 1	percent states tax people taxes united rate says years 10
topic 2	obama says president said barack city 30 country number illegal
topic 3	health year care billion budget spending 000 tax federal plan
topic 4	says texas state voted republican times day senate romney trump

For both combined:

topic 0	says obama president billion said barack clinton federal budget government
topic 1	percent health care states people taxes pay united country tax
topic 2	says house trump did georgia state donald gun bush tax
topic 3	000 state jobs says million year new cut years wisconsin
topic 4	says tax voted republican senate spending security national dollars oil

Although the results in section 4.1 have shown that some phrases are indeed more highly correlated with fake news, our LDA results show that the majority of words in each sentence are found in both real and fake news. There does not appear to be immediately obvious latent structure that distinguishes between fake and real news, although we attempted to vary the number of topics. This is perhaps unsurprising, as circulators of fake news on political sites attempt to make their news as convincing as possible. As LDA assumes that word occurrences in a document are independent, it fails to notice connections between words in a sentence. This suggests that treating words independently, as in a bag-of-ngrams representation, might be insufficient to produce reliable classification

results. This informed our decision to explore methods, such as Parts-of-Speech tagging, that explores underlying syntactic structure within a sentence.

4.3 Classifiers

Classifier	Precision	Recall	Specificity
Bernoulli NB	**0.615**	0.716	0.431
SVM	0.601	**0.817**	0.312
Random Forests	0.593	0.679	0.408
Multi-Layer Perceptron	0.588	0.641	0.431
Decision Tree	0.589	0.616	**0.456**

Table 1: Classifier performances. We provide evaluation metrics for all of our classifiers. The bolded values highlight the best performing classifier for each metric.

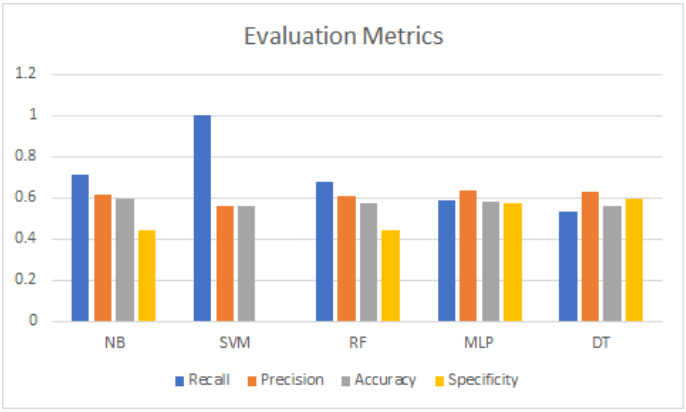

Figure 3: The bar graph shows the evaluation metrics recall, precision, accuracy, and specificity for the five classifiers when feature selection of the 30th percentile is used.

Based on Table 1 and Figure 3, it is clear that the classifiers perform very differently. We are paying special attention to specificity as opposed to recall because we believe false positives are more important than false negatives. It is more detrimental for people to believe false news than it is for people to ignore actual news. Therefore, while the SVM outperformed other classifiers in recall, it did not do as well as any other classifier in specificity. It is interesting to note that the NB, SVM, and RF classifiers had higher recalls than specificity; perhaps they would be better used on problems where false negatives are more important than false positives.

4.4 Feature Selection

The ROC curves will help us determine the optimal feature selection to use in this problem. The feature selection used in this project is based on the top percentile of features ranked by score. The code was tested with percentile = 10, 20, 30, and 40 in order to determine the most ideal feature selection. We looked at area under the ROC curves as well as computation speed, described in the section below, to determine that the 30th percentile produces the best results at a reasonably fast pace.

Other Types of Feature Selection: Feature Selection by f_classif

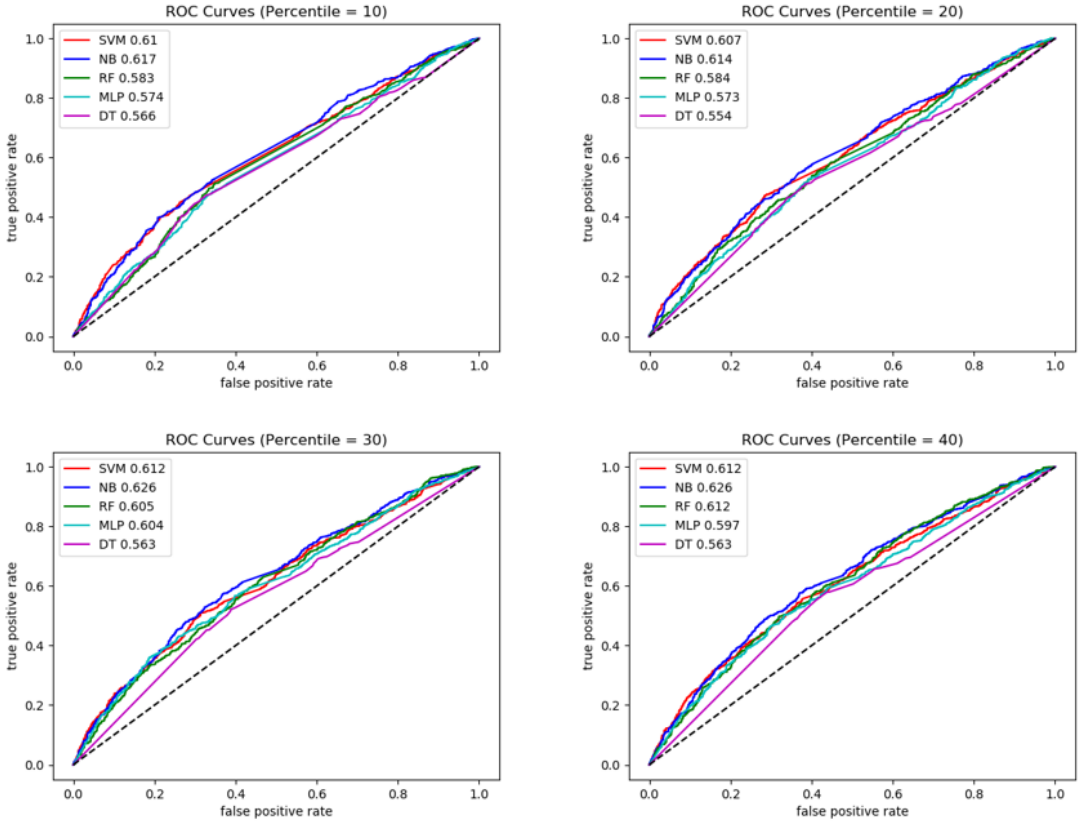

Figure 4: These are the ROC curves for the five classifiers as the number of features selected (by percentile) increase from 10 to 40.

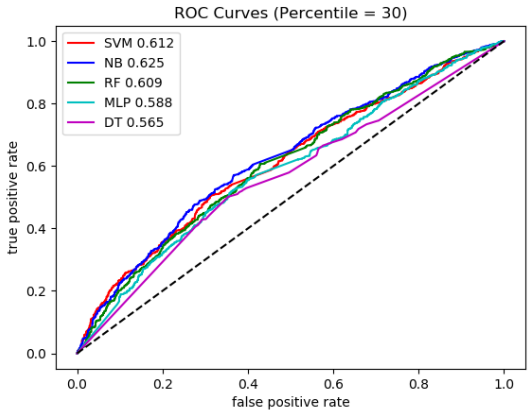

Figure 5: These are the ROC curves for the five classifiers with percentile=30 based on sklearn's f_classif feature selection.

The feature selection was run with the top 30th percentile, this time with f_classif instead of chi2 like the feature selection done above. The f_classif feature selection ranks the features based on the ANOVA F-values. It is interesting to see that the f_classif and chi2 scoring functions produce slightly different results (using area under ROC cuvrve as the metric) for majority of the classifiers

except for MLP, where the difference is quite significant. This is probably due to the fact that F-Test captures linear relationships only, and MLP is a nonlinear classifier.

4.5 Computational Speed Analysis

The graph below shows the total computation speed for training and testing the classifiers on the data for the five classifiers on the four different percentile values tested:

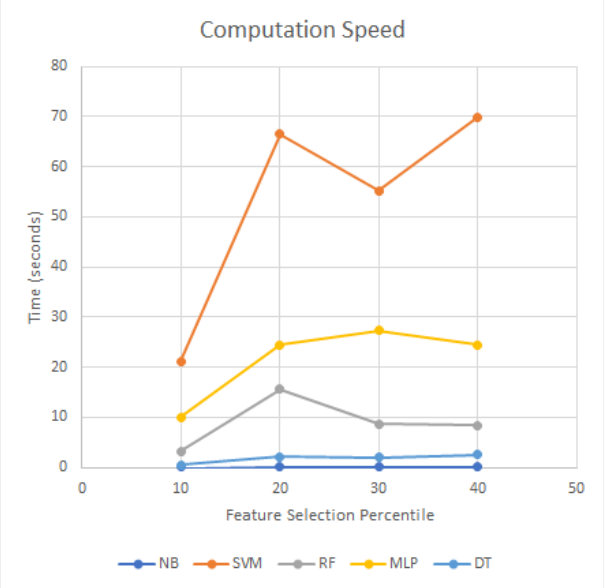

Figure 6: The graph shows the computation speed in seconds as the feature selection percentile range is increased from 10 to 40. The graph includes the timing analysis for all five classifiers.

The SVM Classifier takes significantly longer, while the Naive Bayes and Decision Tree classifiers take significantly shorter. Oftentimes, researchers have to make the decision of whether to trade off computational speed for accuracy. However, in this situation, it seems like the more efficient algorithms are out-performing some of the more complicated ones. For example, the more efficient classifier Decision Trees outperformed the SVM classifier in precision, accuracy, and specificity. Additionally, this graph shows the relationship between increasing the number of features selected and the computational speed. Generally, increasing the number of features increases the total computation time. As discussed above, the 30th percentile and 40th percentile selections do not create significant differences in evaluation metrics. Plus, there is no significant increase in time from the 20th to 30th percentile, so it is justified to choose 30th percentile over the 20th and 40th percentiles.

4.6 Cross Validation

We attempted grid search cross validation on three of our classifiers (Bernoulli Naive Bayes, SVM, Random Forests, Multi-Layer Perceptron), seeking respectively to tune the hyperparameters alpha (additive Laplacian smoothing factor), gamma (kernel coefficient), n_estimators (number of trees in the RF) and alpha (L2 penalty). Even after using feature selection, none of SVM, RF or MLP grid search completed within reasonable time.

As seen below, our cross validated Naive-Bayes classifier did better than the original estimator in recall, but performed worse in terms of precision and specificity. Overall, we feel that the original classifier was stronger. In this case, cross validation did not actually improve our classifier, which could have been a result of various factors such as patterns left unaccounted for within the dataset, less training data available after the training data is split, and many more [12].

	Precision	Recall	Specificity
Without GridSearchCV	0.615	0.716	0.431
With GridSearchCV	0.603	0.755	0.370

Table 2: Classifier performances with and without grid search cross validation.

5 Discussion and Conclusion

5.1 Parts-Of-Speech (POS) tagging and Neural Network model - SpaCy

Classification Recall: 0.706 Precision: 0.605

In comparison to a bag-of-words model of n-grams, which breaks sentences into sequences of words without distinguishing for semantic meaning, this approach allows the model greater consideration of linguistic context. Although it is unsurprising that this model can handle headlines with greater complexity in sentence structure, this model is generally slower than statistical models as it has to consider a greater combination of connections between words in a sentence.

5.1.1 Misclassified Samples

Some samples were marked as real news by the SpaCy model with 99% confidence, although they were in fact fake. An example is 'Says Ohio budget item later signed into law by Gov. John Kasich requires women seeking an abortion to undergo a mandatory vaginal probe.' This sentence has a complicated syntactic structure, which may be difficult for the model to parse. Furthermore, the ridiculousness of phrases like 'mandatory vaginal probe' may not be detected because it is not a commonly used phrase. An example where the model classified the headline as fake when it is in fact real is "'Immigrants are more fertile." and 'When asked about equal pay for women, (Rubios) quote was that it was a waste of time'. The second statement could have been classified as false due to the extreme nature of the phrase "waste of time", which is more commonly found in fake news that is intended to incite emotional outrage.

5.2 Conclusion and Further Extensions

With exploratory data analysis using topic modelling, we discovered that many of the phrases found in real and fake news are largely similar. Therefore, a bag of ngrams representation is generally insufficient to produce results with a level of confidence sufficient for industry use. We find that of the statistical models, SVM and NB perform the best in recall and precision respectively. The SpaCy model, utilising Parts-of-Speech tagging on a neural network based model is better equipped to deal with complicated linguistic structures. We found that models tend to misclassify headlines when real news uses language that is more extreme, or if the more ridiculous part of a fake headline is a less commonly used phrase.

Given more time and resources, we would like to see how our trained classifiers perform on new political headlines. As trends in political news have short turnover rates (especially during election seasons), we would expect that fake news classifiers would have to be updated with newly trending topics frequently. It would also be beneficial to see how additional input like the source of the article and the number of comments differs between fake and real news. This information can then be weighted in when the classifier assigns labels to samples. Furthermore, it is interesting to note that not all fake news are the same - some are just blatantly false, while others could be satire, or exaggerated statistics. It would be valuable to classify fake news as each of these categories, as it is more likely for a model to misclassify satire than for it to misclassify an extreme opinion.

Application fields such as search engines or social media platforms could utilise a fake news detector to flag certain headlines or posts as possibly fake. If the classifier is highly confident of the news being fake, it could completely remove the article from being viewed. Articles that lie more on the classification margin can instead be tagged as "possibly fake" so that the user can exercise greater caution when reading it.

Acknowledgements

Throughout this project, we used and modified old code from the other assignments, as well as precept code. We used code modified from SpaCy's example page [5], which they stated was free for modification. Methods from SpaCy and SciKit Learn's libraries were used [4] [11].

References

[1] Patro, S. (2019, January 21). Fake News Detection Dataset. Retrieved from https://www.kaggle.com/ksaivenketpatro/fake-news-detection-dataset/activity

[2] Fact-checking U.S. politics. (n.d.). Retrieved from https://www.politifact.com/

[3] Vosoughi, Soroush, et al. The Spread of True and False News Online. Science, vol. 359, no. 6380, Mar. 2018, pp. 114651. science.sciencemag.org, doi:10.1126/science.aap9559.

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[5] Examples spaCy Usage Documentation. (n.d.). Retrieved from https://spacy.io/usage/examples

[6] Oshikawa, R., Qian, J., & Wang, W.Y. (2018). A Survey on Natural Language Processing for Fake News Detection. CoRR, abs/1811.00770.

[7] Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg. Chicago

[8] Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003). Random forest: a classification and regression tool for compound classification and QSAR modeling. Journal of chemical information and computer sciences, 43(6), 1947-1958.

[9] Ruchansky, N., Seo, S., & Liu, Y. (2017, November). Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 797-806). ACM.

[10] Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P. (1992). A practical part-of-speech tagger. In Third Conference on Applied Natural Language Processing. Chicago

[11] SpaCy Industrial-strength Natural Language Processing in Python. (n.d.). Retrieved from https://spacy.io/

[12] Schimtz, Martin. When Cross Validation Fails. Towards data science, 2017. Retrieved from https://towardsdatascience.com/when-cross-validation-fails-9bd5a57f07b5/