Fake News Detection

CS 229 | Final Project | Team ID: 621

Ayush Singhania (ayushs@stanford.edu) | Devyani Choudhary (devyani@stanford.edu) | Sohan Mone (sohanm@stanford.edu)

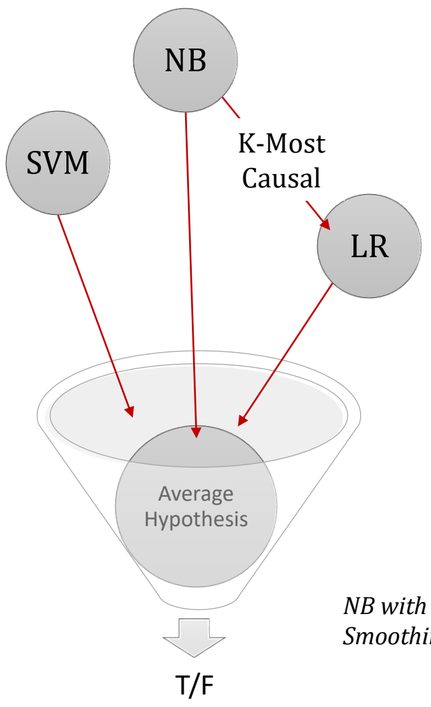
Predicting

Fake news, or maliciously-fabricated media, has taken a central role in American political discourse. Through the development of two models, we were able to generate a system that can discriminate between "fake" and "true" news articles with an 83% accuracy.

Data

- **Data Collection** - Pre-labeled data for this project was obtained from Kaggle. Data marked as "fake" has been identified by the *BS Detector* extension for Google Chrome. For our final model, 4,083 fake and 8,070 true articles were used. The features used are the body (main text) of the article, and the article title. Each word of the article is a token.
- **Natural Language Processing** - NLTK in Python was used to identify, count and consolidate tokens within the obtained data sets. To combine tokens with similar root words, a lemmatizer was used.
- **Data Purification** - The number of total tokens considered was considerably reduced by eliminating stop words (such as "as" "is" "the" etc.. that are not believed to be causal on the article classification. To further purify the data, the extracted data was manually purged of any unreadable or non-English characters or boiler-plate article headings.

Models



- **Average-Hypothesis**- Our average hypothesis model combines Naïve Bayes, Linear Regression and SVM by averaging the output probabilities obtained from each model. The aim of averaging is to obtain a model that is less susceptible to overfitting. Naïve Bayes (with Laplace smoothing) and SVM was ran using all 5,078 tokens, while Linear Regression was performed using only the 20 most causal tokens. The SVM algorithm uses a second-order Gauss kernel.
- **Neural Network**- Our two-layer neural network operates on the 2500 tokens identified to be most causal to a source's classification. The middle neurons use *ReLU* activation, while the final neuron uses s*igmoid* activation.

$$ReLU = f(z) = \max(0, z)$$

$$Sigmoid = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

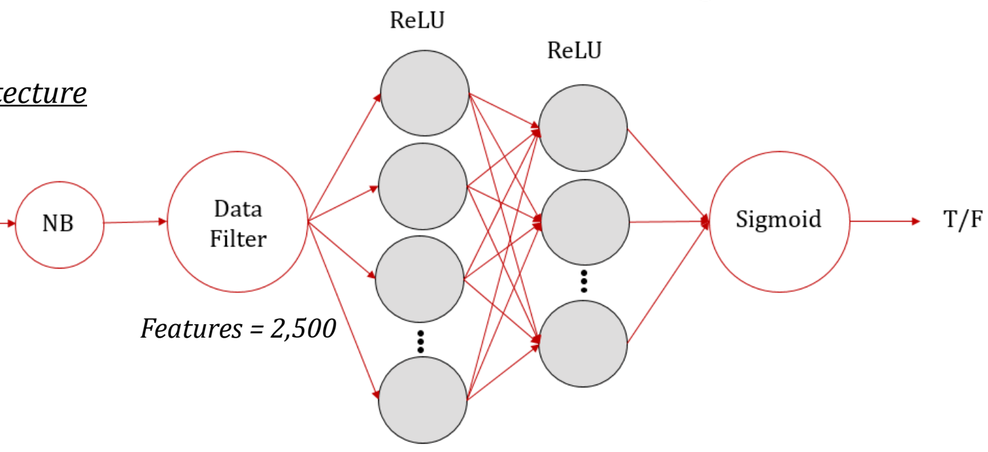
$$J_B = CrossEntropy = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)} \log(\hat{y}^{(i)})$$

NB with Laplace Smoothing:

$$\left\{ \begin{aligned} \phi_{j|y=1} &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} + 2} \\ \phi_{j|y=0} &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} + 2} \end{aligned} \right\}$$

Neural-Network Architecture

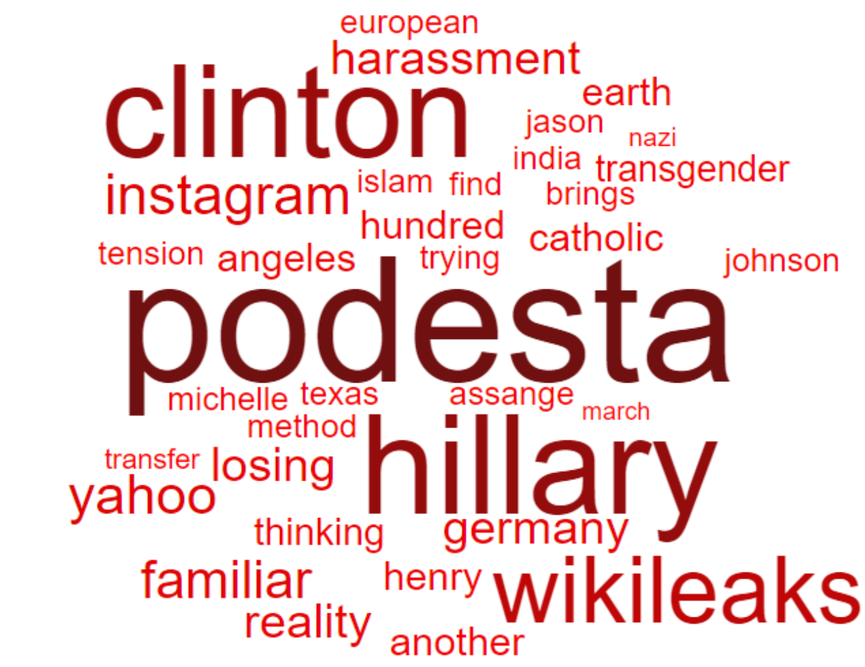
Samples = 12,161



Features = 2,500

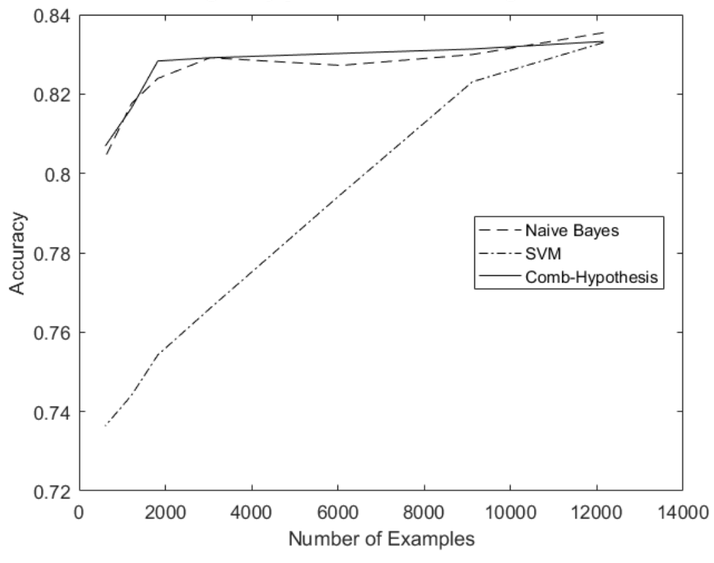
Validation and Discussion

- **Feature Selection** - To investigate our initial assumptions, the model was tested using the tokens from the body + title, from the only the body and from only the title. From these results, we found that including both the body and the title was optimal.
- **Short Circuiting** - Initially, our model was vulnerable to a phenomenon called "short-circuiting". The tokens that caused short-circuiting were found and filtered using Naïve Bayes to generate the *K-most-indicative* token set. The tokens shown in red have the highest *posterior probability* of being in a "fake news" article, while the tokens shown in green have the highest posterior probability of being in "true news". The size of the token's font corresponds to its causality.
- **Data Set Size** - To investigate the influence of the number of training examples on the performance of our model, a learning curve was generated for our average-hypothesis model.
- **Discussion** - From this analysis, we learned that the most frequent tokens are not the most causal ones. The existence of short-circuiting tokens underscores the need to check a ML model with common sense.

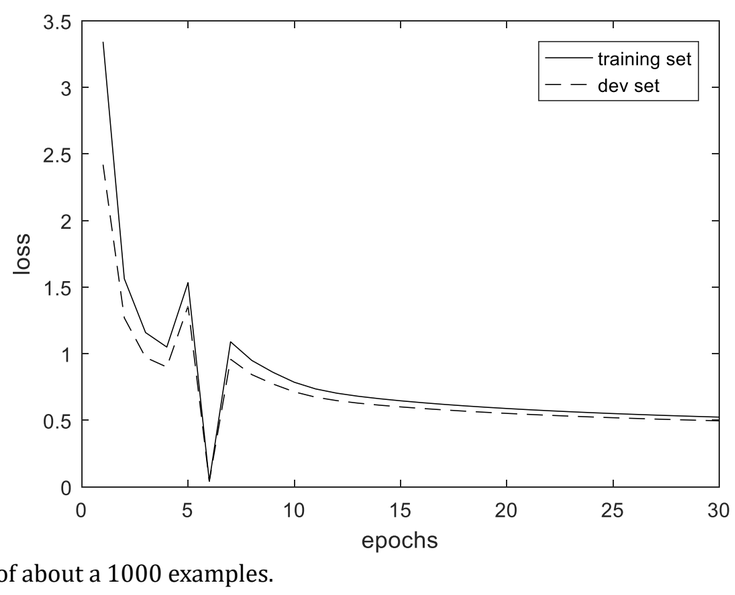


Causality Map of Fake Tokens

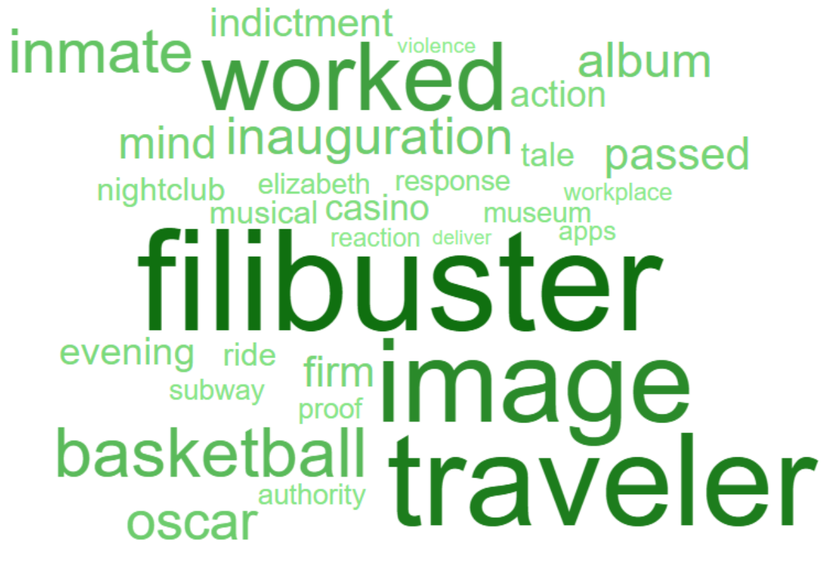
Average-Hypothesis Learning Curve



*Neural-Network Loss Curve ***



**NOTE: The neural-network loss curve is from a smaller data set of about a 1000 examples.



Causality Map of True Tokens

Results and Future Inquiry

- **Accuracy**- Using the Average-Hypothesis method, the accuracy for final model was found to be **83%.** In the future, we will look at using a different SVM kernel.
- At this point, we don't have accuracy results for the full data set using the Neural Network. We will try out different network depths and activation functions.

	Sub Model	Naïve Bayes	SVM	Linear Regression	Averaged-Hypothesis
Accuracy	Body + Title	0.8256	0.8239	0.9899	0.8308
	Body	0.8278	0.8258	0.9953	0.8272
	Title	0.6805	0.6624	0.9956	0.6659

References

Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

Ng, Boneh, (2017), *CS 229: Machine Learning*. Course Material.