Identifying Tweets Written by Russian Troll Accounts

Ethan Brown, Brendan Edelson, Elijah Taylor

esbrown@stanford.edu, bedelson@stanford.edu, elijaht@stanford.edu

Predicting

In the wake of Department of Justice investigations and general public weariness towards cybersecurity, we decided to investigate how deep learning may help in identifying Russian "Twitter bots". Given the text of a tweet, our neural network outputs a binary classification (Russian Bot / Not a Russian Bot).

Data

- 200,000 Russian Bot tweets (ground truth)
  - Released by NBC for public analysis
- Over 1 million politically-themed tweets from the 2016 election season (assumed not Russian bots)
  - Collected through a Harvard research project

Features

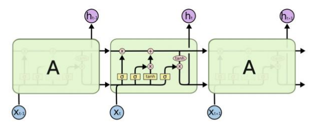
GloVe Vectors

trump	is	loud	😩
$\begin{pmatrix} .048 \\ .621 \\ \dots \end{pmatrix}$	$\begin{pmatrix} .613 \\ .230 \\ \dots \end{pmatrix}$	$\begin{pmatrix} .398 \\ .077 \\ \dots \end{pmatrix}$	$\begin{pmatrix} .005 \\ .094 \\ \dots \end{pmatrix}$

- 200 dimensional feature vectors trained on a Twitter corpus
- Used word embeddings to convert word indexes to GloVe vectors

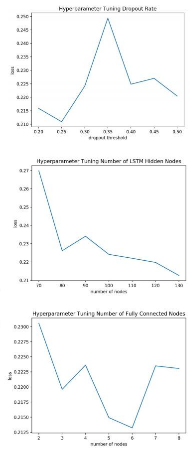
Models

LSTM Hyperparameter	Value
GloVe Embedding Dimensions	200
LSTM Hidden Nodes	130
Fully Connected Nodes	6
Dropout Rate	.25
Epochs	10
Minibatch Size	128



Binary Cross-Entropy Loss

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1-y_i) \cdot log(1-\hat{y}_i)$$



Results

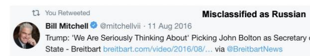
Model	# Train Examples	# Test Examples	Train Accuracy	Test Accuracy
3-Layer NN w/ 25D Glove vectors	35,000	5,000	68.12%	62.15%
LSTM w/o Fully Connected Layer	1.1 Million	12,000	97.45%	94.93%
LSTM with Fully Connected Layer*	1.1 Million	12,000	96.69%	95.07%

*Final model

Discussion

Better than expected results!!

- NLTK tokenizer
- Twitter corpus trained GloVe vectors
- Fully connected layer reduces overfitting



Confusion Matrix

$$\begin{bmatrix} 9843 & 153 \\ 161 & 1843 \end{bmatrix}$$

Correct Classification Rates

- Non-Russian: 98.5% (True Positive)
- Russian: 91.97% (True Negative)

Future

- Further analyze misclassified examples to find trends
- Implement other features into the model (e.g. time posted, user account data, etc.)
- Look into creating a GAN to simulate Russian tweets
- Classifying tweets/users with probabilities of various political biases using a softmax output

References

[1] Bird, S., Loper E. and Klein E. (2009), Natural Language Processing with Python. O'Reilly Media Inc.
[2] François, C. (2015). keras. [online] Available at: https://github.com/fchollet/keras [Accessed 9 Jun. 2018].
[3] Hunter, J. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science \ \& Engineering, 9(3), pp.90-95.
[4] Kaggle.com. (2018). Russian Troll Tweets | Kaggle. [online] Available at: https://www.kaggle.com/vikasg/russian-troll-tweets [Accessed 8 Jun. 2018].
[5] Littman, J., Wrubel, L. and Kerchner, D. (2016). 2016 United States Presidential Election Tweet Ids. [online] Dataverse.harvard.edu. Available at: https://doi.org/10.7910/DVN/PDI7IN [Accessed 7 Jun. 2018].
[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825-2830.
[7] Pennington, J., Socher, R. and Manning, C. (2014). GloVe: Global Vectors for Word Representation. [online] Nlp.stanford.edu. Available at: https://nlp.stanford.edu/projects/glove/ [Accessed 8 Jun. 2018].
[8] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.