LSTM For Identifying Russian Political Troll-Bots On Twitter

Brandon Cummings, Tako Hisada, Cameron Kennedy
University of California, Berkeley, W266 Natural Language Processing with Deep Learning
brandon.cummings@berkeley.edu, thisada@berkeley.edu, camkennedy@berkeley.edu

Abstract

The Internet Research Agency (IRA) has been assessed by the U.S. Intelligence Community to be a part of a Russian state-run effort to influence elections and to push political disinformation agendas in many different election cycles across the globe. In 2017, Twitter disabled and handed over 3,814 accounts to the U.S. Congress that were deemed associated to the IRA and accused of "malicious activity" in the 2016 U.S. presidential elections. In this paper, we propose use of a contextual long short-term memory (LSTM) deep neural network to identify tweets and accounts associated to the IRA. Our dataset is comprised of roughly 200,000 tweets from 394 of the "malicious" accounts, as well as roughly 2.8 million genuine account tweets from 3,474 accounts. Our main contribution is the ability to classify IRA "malicious" accounts from a single tweet. Our baseline model trained with tweet text data has a classification accuracy of 90%.

1 Introduction

The United States (U.S.) is going through a unique transition, learning the hard way the impact that social media and digital advertising can have on a democracy when used with malicious intent. In the 2016 U.S. presidential election, it has now been confirmed that the Internet Research Agency (IRA), along with other Russian entities, were invested in swaying the public opinion with disinformation campaigns that promoted divisive propaganda **[1]**. In 2017, the largest social media platforms in the U.S. were called before congress **[2]** to report on advertising spend and behavior of malicious accounts from foreign entities, where initial uncoverings found 10 million users to have been influenced by these malicious accounts. The IRA has been identified as a major contributor to the disinformation campaign on the 2016 U.S. presidential election and ongoing manipulation of the U.S. political environment **[2,6]**. In 2018, the Mueller investigation brought charges to different Russian agencies and persons on two separate occasions, February 16th and July 13th, for U.S. election interference, with a total of 32 individuals and 3 entities **[4,5]**.

In 2017, Twitter disabled and handed over 3,814 accounts to the U.S. Congress that were deemed associated to the IRA and accused of "malicious activity" in the 2016 U.S. presidential elections. After handing over the identified malicious accounts, Twitter removed the 3,814 accounts and roughly 3 million tweets. The news organization NBC was able to capture metadata from 394 of these malicious accounts and 200,000 tweets generated by the 394 accounts, and released this data to the public **[6]**.

The motivation for this paper is simple, be able to accurately identify tweets from the IRA associated with disinformation campaigns aimed at the U.S. political environment.

There are multiple research papers exploring ways to identify trolls on an account and individual tweet level of classification, but none so far that we found accurately identify tweet level classification of malicious tweets associated to the IRA. Using the available 394 malicious account metadata and 200,000 tweets, we aim to answer the following question:

1.1 Research Question

Is it possible to identify an IRA tweet or account from the twitter disinformation campaign of the 2016 U.S. Presidential Election?

1.2 Impact of work

The ability to accurately identify malicious tweets from the IRA aimed at the U.S. political environment would have large practical applications for building trust with the major social media platforms and public information in general. To combat malicious accounts and trolls, Twitter is reported to be deleting close to 1 million malicious accounts/trolls per day **[12],** the work in this paper could aid in this search and deletion of malicious accounts that have not yet been identified.

The U.S. government would also be able to leverage the identification of malicious accounts associated with the IRA to publicly expose the malicious intent of the divisive rhetoric and hold the IRA accountable for its actions. Expanding the work from this paper, the U.S. government could then advise other foreign governments on how to prevent election interference from the IRA.

1.3 What this paper is not

This paper is not an attempt to aid any political affiliation or political agenda, this is aimed at identifying malicious Twitter activity on the U.S. political environment from the IRA.

2 Dataset

Our dataset is a combination of genuine account and tweets are from the dataset used in **[11]**, and the malicious accounts data is from the the NBC News article **[6]**.

Dataset	**# of Accounts**	**# of Tweets**
Genuine Accounts	3,475	2,799,999
Malicious Accounts	394	203,482

Along with the text from each tweet to train our classification, we also included metadata from each tweet and account information. Here are the features we used:

Tweet Data

1) **Text** - the actual tweet text
2) **Retweet Count** - number of times a tweet has been retweeted
3) **Favorite Count -** number of other users that favorited the tweet
4) **Number of Hashtags** - number of hashtags referenced in a tweet
5) **Number of URLs -** number of URLs referenced in a tweet
6) **Number of Mentions -** number other users' handles in the tweet text
7) **Tweet Length -** number of tokens in tweet (project derived)

User Data

1) **Statuses Count -** total number of tweets and retweets from this account
2) **Followers Count -** total number of other accounts following this account
3) **Friends Count -** total number of other accounts this account is following
4) **Favourites Count -** total number of other tweets this account has favourited
5) **Listed Count -** total number of public list this account is on

3 Background

To successfully establish a method to detect the trolls on Twitter by the IRA, we first looked for papers that discussed troll detection. We wanted to replicate and verify the outcomes of the studies; therefore, we needed to find papers with public datasets. We eventually found the paper *Deep Neural Networks for Bot Detection* **[5]** in which the authors discussed utilizing an LSTM architecture for detecting (non-IRA) trolls from humans at individual tweet-level with high accuracy. The authors had made the datasets including original tweets publicly available which we decided to utilize for our own study.

Since we are interested in identifying not just any trolls but specifically the ones maintained by the IRA, we needed to find a dataset that contains labeled tweets by IRA trolls. We accomplished this by incorporating a dataset published by NBC News **[6]** after Twitter took down Russia-linked accounts which were identified to have had engaged in

malicious activities during the 2016 U.S. presidential election. By replacing the troll data in the dataset obtained from the Kudugunta and Ferrara study with the IRA troll data from the NBC News article, we were able to prepare a dataset specifically geared towards detecting IRA trolls.

As the original paper by Kudugunta and Ferrara is not a peer-reviewed paper and the NBC News article not being an academic paper, we decided to augment our study by exploring and incorporating methodologies and results from other well-cited papers discussing the topic of troll detection.

4 Methods

We created an LSTM network, a special variant of Recurrent Neural Networks (RNNs), in modeling the patterns of the IRA troll tweets we are interested in identifying. As trolls become increasingly more sophisticated, the state-of-the-art techniques from NLP using textual content have been proven ineffective against more advanced social trolls **[5]** due to their inability to memorize large state sequences which is critical in comprehending contexts. LSTMs overcome this limitation by not requiring the number of states to analyze in the behavioral model sequence to be predefined **[10]**. All models incorporate variable-length sequences fed to the LSTM. A diagram of our full model is shown in Fig. 1 below.

Similarly to what Kudugunta and Ferrara did in their study, we pre-processed the tweet and account data before applying the GloVE algorithm in creating embeddings. This involved standardizing the data coming from 2 different sources, the genuine tweet data and the Russian troll tweets from the NBC News article **[6]**, then applying tokenization and canonicalization. For the standardizing portion of the pre-processing, we concatenated the 2 tweet datasets then converted NAs to 0s in 4 out of the 5 metadata columns of the IRA data, because the genuine tweets did not contain NAs. Although we recognize changing NAs to 0 reduces fidelity of the original data, we didn't want to give our ML algorithms any 'unfair advantage' to learn that every tweet with an NA value must be a IRA troll, since only the IRA troll tweets contain NAs. We are also comfortable with this decision because nearly all of the genuine tweets contain values of 0 for these same fields. We also converted hashtags, URLs and mention strings found in the IRA troll dataset to counts because the genuine dataset only contained the counts. The tokenization procedure involves converting everything to lowercase, translating entities such as hashtags, URLs, numbers, user mentions and emojis to tags such as "<hashtag>" and "<heart>" and adding some semantic information in the form of tags such as "<allcaps>".

Next, we used a pre-trained set of Global Vectors for Word Representation (GloVE) designed for Twitter data **[13]** in transforming the labeled tweets we obtained into an appropriate embedding we can use in our LSTM modeling. GloVE is a global log-bilinear regression model that (uses) global matrix factorization and local context window methods to effectively learn the substructure of natural language, by training on word co-occurrence **[10]**. We will defer to the original GloVE website **[13]** as well as Kudugunta and Ferrara's paper **[5]** for details of the embedding procedure required in preprocessing Twitter data.

Unique to this LSTM model is the incorporation of metadata (e.g., number of retweets, number of hashtags, etc.) after the LSTM layer of the model but before the softmax output that produces the final prediction of a genuine or IRA troll tweet. Adding this metadata layer is accomplished by concatenating the metadata onto the final state of the LSTM.

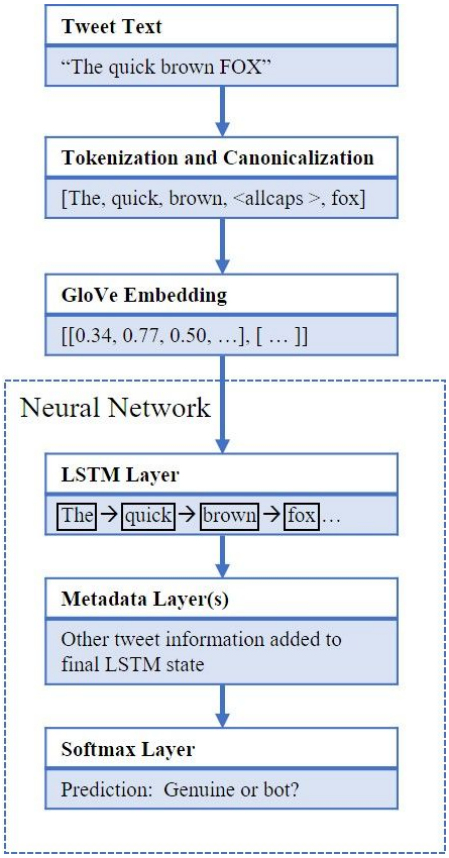
Baseline model

Model 1, the initial baseline LSTM model, combines only a subset of the genuine tweets (5%, 140K tweets) plus all 200K IRA tweets for a total of 342,837 out of the ~3M tweets. It also only uses the tweet text (and associated GloVe embeddings), excluding any metadata available in dataset.

Full-data LSTM models

The full data models incorporate the metadata (scaled and centered) associated with tweets and users such as number of hashtags, URLs and mentions to further improve the accuracy of the model. Model 2 incorporates 11 metadata fields, and Model 3 incorporates 7 metadata fields.

Figure 1: Full Model Diagram



5 Results and Error Analysis

The research team chose accuracy as its primary metric. Other measures observed include precision, recall, F1 score, and ROC AUC, and although these were highly correlated with model accuracy, the team avoided any significant analysis using these measures to prevent taking sides on whether desirable behavior was to identify as many troll tweets as possible at the sacrifice of misclassifying genuine tweets (high recall), or if it's better to err on the side of classifying tweets as genuine at the expense of allowing some troll tweets (high precision). The three models scored as follows:

Model	**Accuracy**
1. Baseline: Tweet Text Only	89.6%
2. Tweet Text + All Metadata	99.2%
3. Tweet Text + Some Metadata	96.2%

To analyze errors, model results were classified into four categories: true positives, true negatives, false positives, and false negatives, and all of the subsequent error analysis was classified into these categories.

The team approached error analysis beginning with the baseline model (Model 1, text only with no metadata) and its accuracy of 89.6%, which aligns with the approach in *Deep Neural Networks for Bot Detection* **[5]**.

Error Analysis: Text

Manual inspection of tweet text and tokens revealed a few key findings. Both false positives and false negatives seemed to have a high number of URLs. They also contained several concatenated words, such as 'weirdthingstobuyonline', both as standalone words and as hashtag phrases that are usually unknown to the GloVe model. Retweets were also found to confuse the model more than original tweets, and they were removed in Models 2 and 3, allowing the project to focus on original, IRA-created content. This change alone improved baseline accuracy to 95.3%. Finally, the text of many false positives actually appeared positive to the research team, and false negatives actually appeared negative, suggesting that some tweets may be nearly impossible to classify.

Perhaps the biggest observation was that Model 2 was performing exceptionally well, leading the research team to consider that it might be learning key features present in one class but not the other. Analyzing the prevalence of tokens and combinations of tokens in each class uncovered 7 tokens (<allcaps>, <user>, :, rt, <url>, <hashtag>, and '…') that were much more common in troll tweets. Of these, the biggest was the string 'rt' which was found in 66% of troll tweets compared to 1% of genuine ones. This 66:1 difference (and others like it) undoubtedly helped the classifier, however; in this example, it still left 35% of troll tweets without this aid, lending credibility to the model since it also correctly classifies the remaining troll tweets. To quantify their impact, alternate trials of the models having removed these 7 tokens in the data set yielded the following results: Model 1 accuracy dropped by 4.1%pts to 85.5%; Model 2 remained unchanged; and

Model 3 dropped by 3.5%pts to 92.7%. The modest performance reduction in Models 1 and 3 shows that while these 7 tokens help the models, they do not have undue influence on them, but the lack of change in Model 2 suggests that something else is driving its performance (see *Error Analysis: Non-Text* section below). Finally, notably, the 'rt' string was frequently found in tweets whose metadata tags indicated they were not retweets, suggesting that troll tweets might have used methods that manually inserted this text.

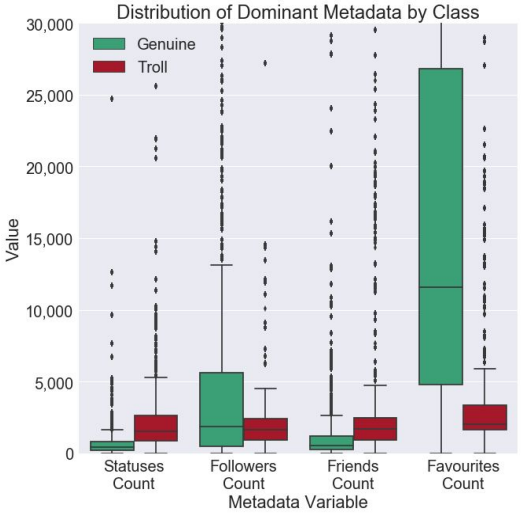
The next area of text analysis was unknown token identification. Inspecting a list of the top tokens the GloVe model didn't recognize immediately revealed the custom tokenizer was frequently not separating punctuation from text, which the research team fixed in all its models. Concatenated words were also common unknown tokens, as discussed above.

The final main category of error analysis came from listing unique words in false positives / negatives (effectively removing common stop words, such as 'the', 'a', 'to', etc.). Here, false negatives frequently contained politically oriented and / or concatenated words, e.g., 'trumpforpresident' was the most common. False positives had no discernable pattern in this category.

Error Analysis: Non-Text

Adding tweet metadata (and associated hidden layers after the LSTM) to the text-only baseline was the single biggest improvement to the baseline model, as seen in Model 2's very high accuracy. Careful inspection and analysis of this metadata was performed to test the extent to which any direct patterns enabled the model to perform with artificially high accuracy. Although this issue was not present in the tweet metadata fields (over 75% of all tweets, genuine and troll, had values of 0), it was glaringly present in 4 of the 5 user metadata fields: statuses_count, followers_count, friends_count, and favourites_count. Boxplots of these fields' values showed drastic differences between genuine and troll tweets (see Fig. 2), and a trial run of the model with no text and only these 4 features performed nearly identically (99.0% accuracy) to Model 2, revealing that these fields were essentially feeding an 'answer key' to the model, causing the tweet text to make no contribution to the outcome and effectively eliminating the need for natural language processing.

Figure 2: Dominant Metadata Fields



Because the project aims to study the effects of text, the team created Model 3, which removes the 4 dominant metadata fields and keeps the remaining 7. Running the same test as above, with no text on only the 7 remaining fields yielded a below-baseline 88.8% accuracy, allowing the team to conclude that Model 3 effectively incorporates contributions from both the text and the 7 metadata fields.

In addition to the metadata improvements, the team also improved the model by adding ReLU activations to the metadata layers. This addition fixed a problem -- incidentally, that adjusting learning rate could not; thus ruling out divergence from overly aggressive learning -- where the accuracy had plateaued but was changing considerably between epochs (Model 2 accuracy: ~95%±2.0% before ReLU; ~99%±0.2% after), and became more prevalent when adding additional network layers. The team hypothesizes that the ReLU layers overcame a vanishing gradient descent problem.

Full Data Performance

Near the end of the project, the team fully re-architected the model to process the complete data set of 2.8 million genuine tweets. The primary change was moving away from loading all the embeddings for all the tokens in a single pandas dataframe to instead loading the embeddings in small

batches during the TensorFlow batch creation process, thus significantly reducing memory (RAM) required. Although the bulk of the project and error analysis focused on analyzing results with only 5% of genuine tweets, the team generated accuracy for the models using the full data as follows:

- Model 1: 96.3% accuracy
- Model 2: 99.7% accuracy
- Model 3: 98.4% accuracy

These results represent solid improvements compared to the partial data set, and align with expectations that more training data would improve performance. The team also noted that the model performed well despite the imbalance issues the additional data presented (~14:1 genuine:troll).

6 Conclusions and Future Considerations

This research supports the conclusion that the technique used to classify spam tweets in *Neural Networks for Bot Detection* **[5]** is also a viable method for classifying IRA tweets. Specifically, adding metadata layers after an LSTM clearly improved model performance, resulting in higher accuracy. The team also identified several opportunities for improvement throughout the project and/or for future work:

1) The data came from two different sources, one for genuine tweets and the other for IRA tweets. The team was unable to obtain this data from a single source; therefore, it potentially introduced 'giveaway' features that might have led to overperformance. Though the team diligently looked for this issue between the two data sets, and even built Model 3 to overcome suspiciously accurate results in Model 2 arising from such 'giveaway' features, ideally, extracting both genuine and troll tweets from the same source would eliminate any potential for this issue.

2) The team observed concatenated words occurring in false positives and negatives, and thus hypothesizes that adding a text splitter for concatenated words, hashtags, and usernames might improve performance. Such a splitter would need to be configured to split both common words and domain-specific words such as political candidate names, along with avoiding splitting uncommon / nonsense words.

3) Having seen a high number of URLs in false positives and negatives, the team hypothesizes that feeding data from those links' websites to the model might also improve performance, though both acquiring and processing the content of these sites would likely be resource intensive.

4) In retrospect, a better architecture for this model would have been to generate the embeddings for the tweets at the time of processing (batch by batch, as opposed to generating them for the entire dataset), thus saving computing resources and allowing model scalability. Although the team eventually made this change, it would have saved time to use this architecture from the start.

5) Oversampling IRA tweets might further improve performance when using the full dataset, and the Synthetic Minority Over-sampling Technique (SMOTE) might be a viable method to accomplish this task.

6) Finally, on July 31, 2018 (10 days before this project's deadline) website FiveThirtyEight released the full set of three million IRA tweets **[14]**. This data did not contain tweet or user metadata, but a logical next step for continuing this work would be to attempt to obtain the metadata for these tweets and incorporate them into the model.

7 Supporting Code

https://github.com/datasci-w266/2018-summer-assignment-1BrandonCummings/tree/final-project/Final-Project

This link contains the python code used to generate the research findings. There are 6 files used for the project, as follows (all file names begin with NBxxa corresponding to the codes here):

1. **NB01**: Loading data and joining user metadata
2. **NB02**: Tokenize, canonicalize, and embed 5% of genuine tweets; save dataframe
3. **NB02b**: Tokenize, canonicalize 100% data, save arrays / lists
4. **NB03**: LSTM for use with 5% of genuine tweets, heavy error analysis
5. **NB03a**: LSTM exploring metadata alone, with no text
6. **NB03b**: LSTM for use with 100% data

References

1) Case 1:18-cr-00032-DLF, US v Russian Entities, 16 Feb 2018, *First Indictment,* https://www.justice.gov/file/1035477/download
2) Romm, "Facebook and Twitter will testify to the U.S. Congress on Russia and the 2016 presidential election", 4 Oct 2017, *Recode.net* https://www.recode.net/2017/10/4/16424514/facebook-google-twitter-testify-house-senate-intelligence-committee-congress-russia-investigation
3) Case 1:18-cr-00032-DLF, US v Russian Entities, 13 July 2018, *Second Indictment,* https://d3i6fh83elv35t.cloudfront.net/static/2018/07/Muellerindictment.pdf
4) Prokop, "All of Robert Mueller's indictments and plea deals in the Russia investigation so far", *Vox.com,* https://www.vox.com/policy-and-politics/2018/2/20/17031772/mueller-indictments-grand-jury
5) Kudugunta, Sneha, and Ferrara. "Deep Neural Networks for Bot Detection." 12 Feb. 2018, arxiv.org/pdf/1802.04289.
6) Popken, Ben. "Twitter Deleted Russian Troll Tweets. So We Published More than 200,000 of Them." *NBCNews.com*, NBCUniversal News Group, 14 Feb. 2018, www.nbcnews.com/tech/social-media/now-available-more-200-000-deleted-russian-troll-tweets-n844731.
7) Liu, Linqing, et al. "Detecting 'Smart' Spammers on Social Network: A Topic Model Approach." *Proceedings of the NAACL Student Research Workshop*, 28 Apr. 2016, doi:10.18653/v1/n16-2007.
8) Volkova, Svitlana, and Bell. "Account Deletion Prediction on RuNet: A Case Study of Suspicious Twitter Accounts Active During the Russian-Ukrainian Crisis." *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, 17 June 2016, doi:10.18653/v1/w16-0801.
9) Chu, Zi, et al. "Who Is Tweeting on Twitter." *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC 10*, 10 Dec. 2010, doi:10.1145/1920261.1920265.
10) Torres, Catania, Garcia and Garino. "An analysis of Recurrent Neural Networks for Botnet detection behavior" *2016 IEEE Biennial Congress of Argentina (ARGENCON)*, 10 Oct. 2016, doi:10.1109/ARGENCON.2016.7585247
11) Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. 2017. "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race". *In Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 963–972.
12) Al-Heeti, "Twitter suspending 1M accounts a day in fight against disinformation, report says", *cnet.com,* https://www.cnet.com/news/twitter-suspending-over-1m-accounts-a-day-in-fight-against-misinformation-report-says/
13) Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation, https://nlp.stanford.edu/pubs/glove.pdf
14) Roeder, Oliver. "Why We're Sharing 3 Million Russian Troll Tweets." *FiveThirtyEight*, 31 July 2018, https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/