Predicting Twitter Trends: How Opaque is the Black Box?

I. Introduction

Ever since hackers, spammers, and politicians discovered the role that social media plays in user engagement, algorithms have been under attack. One particularly vulnerable algorithm is the Twitter trends algorithm. When hashtags, words, phrases, or events begin to trend, they are given front-page space to millions of users across the globe [1]. While this algorithm was intended to simply mirror the network effects of trends within the Twitter platform, it has become a beacon for malicious engagement manipulation.

In 2011, the spam-as-a-service market attacked Twitter trends, censoring public opinions about Russian parliamentary election results [2]. In 2012, politicians running for the presidential election in Mexico hired thousands of citizens to shape public opinion through Twitter trend manipulation [3]. In 2015, the trend algorithm was exploited in an attempt to disengage the public from nuclear negotiations in Iran [4], and also used to drive political opinions of the public in Venezuela [5]. Hackers have discovered how to artificially inflate fake trends (see: trend stuffing [6]) and artificially deflate real trends to manipulate public knowledge and engagement. There even exist secret markets and groups that can be hired to artificially create a Twitter trend for a small price [7].

In the past, Twitter has responded to these attacks by increasingly *making the black box more opaque*. They have added more features and complex weights to determine whether or not something is trending. However, this approach can be problematic, since complex algorithms that lack transparency are hard to audit. Which brings me to the focus of this project, to explore if Twitter's trend algorithm has become too complex for users to understand what is making a topic trend.

Research Question: Can I successfully reverse engineer the Twitter trending algorithm with publicly available data to determine what features and weights Twitter uses to define trends?

II. Methodology

To collect data, I began by creating 2 AWS DynamoDB instances, one that would store a 1% world-wide stream of Tweets, collected from Tweepy's streaming API, and the other that would store an array of all officially trending topics every five minutes, from the Twitter API. I ran the scripts to collect this information intermittently over the course of five days to collect a total of 3,687 official trend arrays, and a total of 1,040,564 tweets. Due to the time and complexity constraints of this project, I chose to focus this study only on trending hashtags, since other trends would require heavy topic modeling. After scraping the data from the databases, cleaning it to omit non-hashtags, empty tweets, and error response codes, and only using tweets that had a corresponding trend array within 10 minutes of the tweet, there was a total of 2671 trend arrays, and a final total of 232,134 hashtags.

For each hashtag, I obtained and/or calculated the following information:

1. The time that the hashtag was tweeted
2. The 10 minute start and stop interval of this hashtag that corresponds to an official trend array from the API
3. The total count of uses of this hashtag from the sample in the last ten minutes before and including this tweet
4. The proportional count of uses of this hashtag from the sample in the last 10 minutes before and including this tweet (proportional to the total amount of hashtags in the last 10 minutes)
5. The mean over the last hour of all ten minute proportional counts of the use of this hashtag from the sample
6. The standard deviation over the last hour of all ten minute proportional counts of the use of the hashtag from the sample
7. The z score for this hashtag (computed from the previous three data points) in the last hour
$$\frac{(proportional\ count\ from\ last\ 10\ minutes - mean\ of\ last\ hour)}{standard\ deviation\ of\ last\ hour}$$
8. A label: 1 if this hashtag was tweeted while it was on the official trend list at the time of the tweet, 0 if this hashtag was tweeted and was not on the official trend list at the time of the tweet

Other features that have been known to contribute to Twitter trends that I could have computed for each hashtag given more time would be trend coverage, potential coverage, reputation, and transmission [8].

In the dataset, originally there was an average of about 10.8 trending hashtags tweeted that existed in any 10-minute window, and an average of about 243.4 non-trending hashtags tweeted in any 10-minute window. Due to this imbalance, I chose to resample the non-trending hashtags in the dataset so that for every 10-minute window, there was an equal amount of trending versus non-trending hashtags for the training data.

All of these features were then fed into a binary SGD classifier, the classification was either 0 or 1 from the “trending” label. After four rounds of feature engineering, I found an optimal set of features (mean, standard deviation, proportion count, and z scores), that created a hashtag-trend predictor with ~93% accuracy.

III. Results & Discussion

For the SGD classifier, I randomly split my dataset into four datasets: X_train, X_test, y_train, and y_test. Each iteration of the training, I chose to use different sets of features to see which features would create a model with the best performance. In this study, the performance metric was sk-learn’s classification_accuracy score, which is the same as a jaccard index score for classification problems. I first trained the model using all of the features available, which led to an accuracy of about 50%. The final model with the best performance was the one that included only the features that contributed to the z score of a hashtag (including the z score itself), which led to an accuracy of about 93%.

A. Limitations

Overall, 93% accuracy for this classifier was quite better than I predicted would be possible, given only publicly available Twitter data. Though, there were a few limitations and possible errors introduced throughout the methodology of this study. First, I chose to only use hashtags from the sample as opposed to all possible topics. This means that there are likely other features involved in the topic modeling process that would raise the overall accuracy of the classifier that I did not include. As well, I chose to not include the hashtag itself (likely as a word embedding) in the classifier, due to time constraints, which could have raised the accuracy as well.

There was also a discrepancy in the dataset of streamed tweets. For the first few days of streaming, I was collecting 1/100 of the 1% Tweepy stream. Then, I chose to collect 1/10 of the 1% stream for the last few days. This change in the velocity of tweets could have skewed the classifier since the sample size for the first few days for every ten minute interval was much smaller than the sample size for the last few days. This could be especially problematic after resampling the dataset to only include the same number of non-trending hashtags as trending hashtags, since often in the first few days there were just 1-3 trending hashtags tweeted in every 10 minute interval. Finally, since I was unable to continue to use the same EC2 instance to run the scripts for scraping data continuously throughout the data collection process, there are significant gaps in the data collection that likely skewed the z scores for specific hashtags and potentially diminished the overall accuracy of the model.

B. Ethical Considerations

It is important to note some ethical considerations in this study. While I streamed tweets from Twitter over the course of several days, I never explicitly asked for consent from Twitter users to use any of these tweets in my model. Although, given that each hashtag was stripped from its tweetId, and eventually stripped from its context entirely, the privacy of the users who tweeted these hashtags has been mostly preserved. Additionally, I did not check my dataset continuously for tweets that have been deleted from Twitter, to ensure that I delete them in my own dataset (as per section I.C.3 of Twitter's Developer Policy [9]).

IV. Conclusion

Overall, this study aimed to reverse engineer the Twitter trending algorithm in an attempt to "make the black box less opaque". After almost a week of streaming Twitter data, a model was created that successfully classified hashtags as trending or not trending with ~93% accuracy. The features included in this model were the proportional count of the use of that hashtag in the last ten minutes, the mean proportional count and standard deviation of the hashtag in the last hour, and the z score of that hashtag in the last hour. Given more time, potential future work would include creating a classifier for all trending topics, including word embeddings as features, and calculating additional metrics specific to each tweet's user profile that could raise the accuracy of the classifier. For more information about this study, and to run the code yourself, feel free to look at the code on the GitHub repository.

References

[1] https://help.twitter.com/en/using-twitter/twitter-trending-faqs

[2] Thomas, Kurt, et al. “Adapting Social Spam Infrastructure for Political Censorship.” https://www.usenix.org/system/files/conference/leet12/leet12-final13_0.pdf.

[3] Treré, Emiliano. “Opening Governance.” *IDSBulletin*, vol. 47, no. 1, Jan. 2016, pp. 127–139., https://opendocs.ids.ac.uk/opendocs/bitstream/handle/20.500.12413/7697/IDSB_47_1_10.190881968-2016.111.pdf?sequence=1&isAllowed=y.

[4] Najafabadi, Mahdi M. “A Research Agenda for Distributed Hashtag Spoiling: Tails of a Survived Trending Hashtag.” *Dg.o '17 Proceedings of the 18th Annual International Conference on Digital Government Research*, 7 June 2017, doi:10.1145/3085228.3085273.

[5] Forelle, Michelle, et al. “Political Bots and the Manipulation of Public Opinion in Venezuela .” https://arxiv.org/pdf/1507.07109.pdf.

[6] Irani, Danesh, et al. “Study of Trend-Stuffing on Twitter through Text Classification.” *CEAS 2010 - Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, 13 July 2010, https://www.cc.gatech.edu/projects/doi/Papers/DIrani_CEAS_2010.pdf.

[7] Thomas, Kurt, et al. “Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse.” *Proceedings of the 22nd USENIX Security Symposium*, 14 Aug. 2013, https://www.usenix.org/system/files/conference/usenixsecurity13/sec13-paper_thomas.pdf.

[8] Zhang, Yubao, et al. “Twitter Trends Manipulation: A First Look Inside the Security of Twitter Trending.” *IEEE Transactions on Information Forensics and Security*, 2016, pp. 1–1., doi:10.1109/tifs.2016.2604226.

[9] https://developer.twitter.com/en/developer-terms/policy