Graph-Based Methods to Detect Hate Speech Diffusion on Twitter

1st Matthew Beatty
Department of Computer Science
Harvard University
Cambridge, MA
mbeatty@post.harvard.edu

***Abstract*—In this paper, we investigate models to detect the spread of hate speech on Twitter based on its diffusion in the network graph. We experiment with a dataset of 10,000 tweets manually labelled as hate speech or not and show that classification based solely on the sharing graph yields strong F1 scores for our task and high hate speech detection precision. We also highlight the vulnerability of existing textual hate speech detection methods to adversarial attacks and demonstrate that while our methods do not outperform state-of-the-art text models, graph-based models provide robust detection mechanisms and are able to detect instances of hate speech that fool text classifiers. We find that graph convolutional networks produce the strongest hate speech F1 score of 0.58 and find other success with kernel methods. Finally, we also consider the effects of automated bots in the sharing of hate speech content and find they are insignificant in our experiments.**

***Index Terms*—graph classification, graph mining, graph kernels, hate speech, Twitter**

I. Introduction

Online social networks have connected people across the world and made sharing information easier than ever, but reports of active hate groups and online hate speech distributed on the platform trouble observers. The anonymity, immediacy, and global reach of online social networks make the platforms attractive broadcast mediums for hate groups. Several large social networking platforms have identified the issue, but current responses have not solved the problem. In 2018, Twitter representatives suggested that the company's automated detection systems could identify over 40% of content that requires moderation using machine learning techniques but admitted that finding all hate speech using automated detection mechanisms is prohibitively difficult [1], [2].

The classification performance of automated hate speech detection systems has improved dramatically in recent years, but three issues still plague existing methods. Hate speech accounts for a tiny fraction of all shared content on the platforms, leading to a highly-imbalanced classification problem. The second issue is that the subjectivity of defining hate speech makes labelling data for an automated detection system even more difficult. There is no consensus definition of hate speech, and the context differences mean clear-cut definitions are difficult to apply in many scenarios. Finally, another issue with current text-based detection methods is that they can be easily fooled by adversarial attacks. Previous research has shown that trivial manipulations of messages, such as adding positive sentiment words or small typos in hateful terms, can fool systems built to detect toxic language [3] [4]. Current systems also have poor detection of coded language in content, and the ability of simple modifications to impair the detection system's abilities significant undermines the recent performance gains of state-of-the-art systems. Our research is motivated by a desire to uncover new methods of hate speech detection to complete existing detection methods. Previous research has analyzed online users who frequently post hate speech content, but few studies to our knowledge analyze the diffusion of hate speech online and whether diffusion patterns can offer predictive detection features.

Our contributions are: (1) we investigate models to detect the spread of hate speech tweets on Twitter and show that graph convolutional networks yield robust F1 scores for our imbalanced classification task and high hate speech detection precision. (2) We reproduce existing results showing how adversarial attacks can weaken text detection models for hate speech detection and demonstrate how our methods, while not outperforming state-of-the-art models, can complement existing detection methods by identifying instances of hate speech missed by purely textual detection methods. (3) We consider the role of automated accounts or "bots" in the spread of hate speech and find that automated accounts play an insignificant role in the spread of hate speech in our experimental dataset.

II. Related Work

Hate speech detection systems have relied primarily on natural language processing, and recent deep learning research has produced the current state-of-the-art detection classification metrics. Earlier research demonstrated the effectiveness of n-grams and bag-of-words techniques. Specifically, Davidson, et al. show strong classification performance with an F1 score of 0.80 on a multi-class hate speech classification task using word-level n-gram features [5]. Deep learning models, such as recurrent neural networks, have been shown to detect hate speech with high levels of precision, and Badjatiya et al. produce a classifier with 0.93 precision and 0.93 recall overall on a dataset of 16,000 labelled tweets using long short-term memory (LSTM) networks with random embeddings

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

and gradient-boosted decision trees [6]. Yet other researchers have undermined the stated gains of text-based classifiers by demonstrating that simple adversarial attacks can fool the state-of-the-art classification models. Gröndahl, et al. show that appending the word "love" to a comment caused hate speech detection precision to drop by 50% for the detection models from Davidson et al., Badjatiya et al., and Google's Perspective API which is used to detect toxic language online [7]. Oak produces similar results that show the drastic effects of minor textual changes on the effectiveness of automated hate speech detection methods [13].

Some research studies have analyzed patterns in the types of users associated with hate speech content, but they did not look at the content's spread. Ribiero, et al. analyze users who create or share hate speech content on Twitter. The authors find that accounts posting hate speech tweets tend to post more frequently than other accounts and also have strong ties in their 1-neighborhood, suggesting strong homophily among that group of accounts [4]. The authors find that hateful users on Twitter frequently retweet each others content and produce tight communities on the platform. Both Zhong et al. and Waseem and Hovy consider the effect of the number of replies, the replies from followers, and the total number of posts by a user to detect hate speech, but the two studies produce opposite results. Zhong finds no correlation between posting activity and hateful content, while Waseem and Hovy find a correlation [8].

To our knowledge, no extensive research has attempted to analyze the spread of hate speech diffusion on Twitter. Previous research has found that information cascades online carrying negative sentiments fade far more rapidly than their positive sentiment counterparts, and Romero, Meeder, and Kleinberg analyze how diffusion cascades on Twitter differed by topic, such as sharing cascades about politics, sports, or movies. [10] [9]. The 2018 research of Vosoughi, Roy, Aral offers a similar investigation of the spread of true and false rumors online. The authors analyze the diffusion cascades of verified true and false news stories on Twitter from 2006 through 2017 and find that false rumors spread faster and more broadly than their truthful counterparts. The authors also find that automated accounts play a limited role in the spread of false news stories [11].

III. Dataset

We use an existing dataset of labelled hate speech tweets compiled by Founta, et al. as the basis for our experiments [12]. The dataset contains 99,799 English tweets created between March 30, 2017 and April 9, 2017 and sampled from Twitter's public API. The authors crowdsource annotations from CrowdFlower to label each tweet as either hateful, abusive, spam, or normal based on definitions derived from related literature and legal dictionaries [12]. Each tweet in the dataset has annotations from five labelers and require a majority label for the tweet to be included in the dataset. The original dataset contains 53,790 (54%) normal tweets, 14,024 (14%) spam tweets, 27,037 (27%) abusive tweets, and 4,948 (5%) hateful tweets.

Next, we gather each tweet's associated diffusion network through Twitter's public API using the tweet IDs. We use a process for recreating the true retweet paths using the limited information from the Twitter API called time-inferred diffusion, described initially by Goel, et al. and used by Vosoughi, Roy, and Aral in their research on the spread of true and false news on Twitter [14]. The Twitter API does not specify the direct retweet path, meaning all retweets in the API response appear to reshare the original tweet even if they shared the content from a later retweet, so we must use this diffusion inference technique to infer the true sharing patterns using follower relationships and retweet timestamps.

We restrict our gathering to tweets with ten or more retweets, because smaller diffusion network graphs will likely not contain enough information for meaningful analysis. At the time we accessed the Twitter API, 42,872 tweets were unavailable either because users had deleted the tweet or Twitter had banned the account or tweet. This undoubtedly adds elements of bias to our hate speech dataset. We are not able to access those tweets in any way, but we suspect that the most extreme forms of hate speech and hate speech content with the most exposure would be the two categories that Twitter would have acted on most quickly. Unfortunately, without access to the tweet texts we are not able to determine exactly how those tweets are different. The follower relationships of users who have deleted their accounts are also not available, and a small fraction (2%) of accounts in the dataset are deleted. All of these weaken our ability to infer the true retweet graph, but we are still able to gather information for 86% of all retweets of the tweets in the dataset, according to tweet metadata. Some tweets in the original dataset are retweets, and we remove all labelled tweets that are retweets from the dataset so that only original tweets and their spread are classified.

Only 6,924 available tweets from the original dataset meet our minimum retweet requirements, and there are 2,154 "abusive" tweets, 967 "hateful" tweets, and 3,203 "normal" tweets. These are larger proportions of abusive and hateful content than the original dataset, and the cause is a boosted abusive and hateful tweet sample of 20,000 tweets from the original dataset. The authors include 20,000 tweets in the dataset after filtering their sample using sentiment analysis and the presence of hate speech terms to gather tweets, and the tweets from this sample tend to be more accessible and more popular than the rest of the dataset [12]. To counteract this, we resample using the opposite filters as the dataset's boosted hate speech sample to gather more "normal" tweets. We gather tweets from the same period in April 2017 and filter by tweets that have high positive sentiment scores and that do not contain any of the hate speech terms the authors require in their original boosted sampling process.

In the end, our experimental dataset includes 10,074 tweets. 2,154 (21.4%) are labeled as abusive, 967 (9.6%) are labeled as hateful, and 6,953 (69.2%) are labeled as neither (i.e. normal). For our binary classification experiments, we combine hateful and abusive into a ”toxic” class.

Class	**Tweets**
Normal	6,953
Toxic	3,121
Total	10,074

Given our minimum size constraints, most eligible graphs cluster towards the minimum size requirement, but several sharing graphs have more than 500 nodes (retweets) or more than 10 layers of depth. In the end, we have a higher proportion of normal tweets and slightly higher number of hateful tweets than the original dataset and fewer abusive tweets.

IV. Experiments

We conduct three sets of hate speech detection experiments on our dataset: classifying retweet networks, classifying tweet texts, and classifying tweet text while applying adversarial attacks. All experiments are set up as binary classification tasks between normal tweets and ”toxic” tweets, the union of abusive and hateful tweets. In each experiment, we optimize for overall F1 score. For each task, we report the precision, recall, and F1 score for the hate speech class. The stated results are aggregated over the same ten 90-10 test-train splits, and we use GridSearchCV for model hyperparameter tuning where appropriate.

For our diffusion network classification, we test regression methods, kernel-based methods, and graph convolution network models. We run logistic regression with a set of hand-crafted features based on structural properties of the retweet graphs, such as the graph depth, cascade duration, node degree distribution, and assortativity coefficient, using 50 features in total to make predictions. Next, we include a series of graph kernel-based learning models. We use the open-source GraKel package for implementations of kerneling functions. We report results for our classification model using a support vector machine classifier with a precomputed Weisfeiler-Lehman kernel. (We also considered a graph shortest-paths and random walk kernels, but after their poor performance, we exclude those models from the results.) We use deep graph kernels proposed in Yanardag, 2015 as well. The kernel uses a neural language model to learn latent representations of the sub-structures in the graphs, using labeling from the Weisfeiler-Lehman kernel. The last graph-based classification model we experiment with is a graph convolutional network (GCN). GCNs are semi-supervised approach to graph classification, reaching state-of-the-art performance on a number of benchmark graph classification tasks. These are extensions of neural networks applied to capture information specific to graph data structures. We perform training in batches of size 64 for the GCNs with at most 100 epochs.

We compare with an implementation of the logistic regression model from Davidson et al. using word-based n-grams and other text-features (readability metrics, tweet length, etc.). We also implement the methods from Malmasi, Zampieri where the authors use SVMs with different word and character-level skip-gram features. We report results from the strongest classifier from their set, specifically using word-level skip-grams features as model input. We also compare our methods against state-of-the-art deep learning models. We investigate two neural network architectures for the task, specifically convolutional neural networks (CNNs) and long short-term memory networks (LSTMs), as described by Badjativa, et al. The authors experiment with both convolution and recurrent neural net architectures for hate speech detection and find that the recurrent LSTMs have the strongest performance in detection hate speech [6]. As part of our experiments showing the effects of adversarial attacks on existing hate speech detection options, we also implement a few simple adversarial attacks from Grondahl, et al. Specifically, we include three adversarial attacks: inserting typos, removing whitespace, and appending the word “love” to the end of hate speech tweets.

TABLE I
Diffusion Graph Classification Results

Method	Class	Precision	Recall	F1
Logistic Regression	Hateful	0.37	**0.53**	0.33
SVM, WL Kernel	Hateful	0.33	0.12	0.20
SVM, Deep WL Kernel	Hateful	0.39	0.18	0.26
Graph Convolutional Network	Hateful	**0.55**	0.34	**0.42**

V. Results

Table 1 includes results for our graph-based methods for the hate speech classification task. Our diffusion classification methods achieve varying levels of success. The graph convolution network produces the best F1 score of 0.40 for the hate speech class and the highest weighted accuracy of our diffusion network methods. Our kernel-based methods reliably detect normal content but have severe difficulty detection hate speech with very low recall scores. In general, the detection methods can classify the majority class of normal tweets well but struggle to separate the retweet networks for abusive and hateful tweets. Another finding is that graph classification accuracy varies with graph size. Our models like the graph convolution network are able to classify graphs with more nodes and those with higher max depths much more accurately. Larger graphs offer more information to make a prediction, and we also find that larger networks in our dataset are disproportionately those of normal content.

The diffusion networks of hate speech tweets tend to propagate widely and quickly at first but do not spread as deeply as other content. All of the graph-based methods tend to learn to classify hate speech based on the same patterns of reoccurring graphlets. The hate speech diffusion networks include a higher proportion of graphlets that include a single parents node with many outward edges with low weights (corresponding to short retweet times). This suggests dense communities of hateful users who share each others content, and the finding is consistent with the results from [4]. We also see our logistic regression model weighted the same type of features heavily, suggesting that the model identified the same patterns.

TABLE II
HIGHEST FEATURE WEIGHTS IN GRAPH-BASED LOGISTIC REGRESSION MODEL

Ordering	Weight	Feature Name
1	0.0387	Variance of the depth distribution of nodes
2	0.0240	Retweet latencies cascade duration 2nd bin (of 10)
3	0.0201	Skewness of the depth distribution of nodes
4	0.0175	Depth distribution of nodes 2nd bin (of 10)
5	0.0159	Skewness of the distribution of retweet latencies

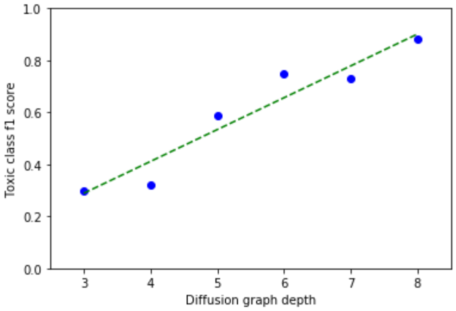

Fig. 1. The average toxic class F1 scores for different cascade depth groups classified by our graph convolutional network with linear trend line.

Next, we compare the performance of existing textual methods on the dataset. Table 2 shows the results of existing text classification implementations on tweets from the dataset. The LSTM classifier has the best performance of our implemented detection methods. It produces the highest F1 scores for all classes and overall accuracy, and the bi-directional LSTM CNN model follows closely in performance. The classifiers using n-gram features do not match the performance of the deep learning architectures but still produce higher F1 scores overall and for the hate speech class than our diffusion network classifiers. While these models outperform our sharing network detection techniques, they still exhibit performance significantly below the stated performance on their original datasets. We attempt to tune hyperparameters to optimize performance, specifically F1 score, but this suggests either there are peculiarities in our dataset that make classification especially difficult or biases in their original datasets that artificially boost performance. In total, there are 116 hateful tweets (12% of total) that were detected by our graph convolution network model that are not detected by any of the text-based methods. We are able to pick out instances of hate speech that do not use obviously derogatory terms. For example tweets such as ”*They have no right be alive in this country if they're jewish. none at all, they're total foreign filth*” or ”*All these hispanic [sic] in LA should be deported Those scum have no right to be here houses should be bulldozed*” do not have severely negative tones or slurs that hate speech text classifiers easily identify. Instead the hateful intentions of the messages requires understanding the meaning of the message.

The convolutional neural networks provide the best classification performances of the implemented text classifiers. They show slight improvements in classifying normal tweets but show dramatic improvements classifying the text of toxic messages. Some of the text classifiers exhibit F1 scores below our graph convolution network model after applying adversarial attacks. The more advanced classifiers using pretrained embeddings have their performance suffer significantly.

TABLE III
SOURCE TWEET TEXT CLASSIFICATION RESULTS

Method	Class	Precision	F1
Logistic Regression	Hateful	0.68	0.56
SVM, n-grams	Hateful	0.60	0.56
LSTM CNN	Hateful	0.74	**0.67**
Bi-directional LSTM CNN	Hateful	**0.81**	0.65

TABLE IV
TOXIC CLASS F1 SCORES WITH ADVERSARIAL ATTACKS

Classifier	Original F1	Typos F1	Whitespace F1	Love F1
Logistic Regression	0.56	**0.35**	0.36	0.39
SVM, n-grams	0.56	**0.33**	0.34	0.43
LSTM	0.67	0.47	0.52	**0.43**
Bi-directional LSTM CNN	0.65	0.51	0.49	**0.39**

The simple addition of the word “love” at the end of each tweet is the most effective adversarial attack for those models, and the n-grams model has the largest performance loss when typos are added or whitespace is removed. Certain examples of toxic language were misclassified by all text classifiers after adversarial attacks by adding ”love” including the following:

- ”*you, or a relative is a pig. That's why I'm telling you to delete your account. You're retarded* ***love***”
- ”*I ain't never had soo much anger for just one person. Like I just wanna beat yo ass bro* ***love***”

Finally, we also consider the role of automated bots in our dataset. Previous studies have considered the role of bots on Twitter's platform and their role in spreading divisive content. We pass each source tweet in our dataset through IUNI's Botometer tool for Twitter bot detection. The tool provides a metric called “Complete Automation Probability” which is the probability that the account and its postings are entirely automated [17]. Few accounts in the dataset have high CAP scores, and only 94 have CAP scores over 0.5. This suggests that the vast majority of accounts are authentic users and that bots are playing a very limited role in the spread of hate speech online. We also find no significant difference in the presence of bots across labels. These results align with the findings from Vosoughi, Roy, and Aral who found that bots played little role in the spread of fake news online [11].

VI. CONCLUSION

We present novel methods for hate speech detection using the aspects of the diffusion of tweets as the sole features for classification. Our detection models offer a potential complement to existing detection methods by detecting a subset of hate speech missed by state-of-the-art text classifiers and a potential backstop against attempts to evade hate speech detection through textual adversarial attacks. Our research

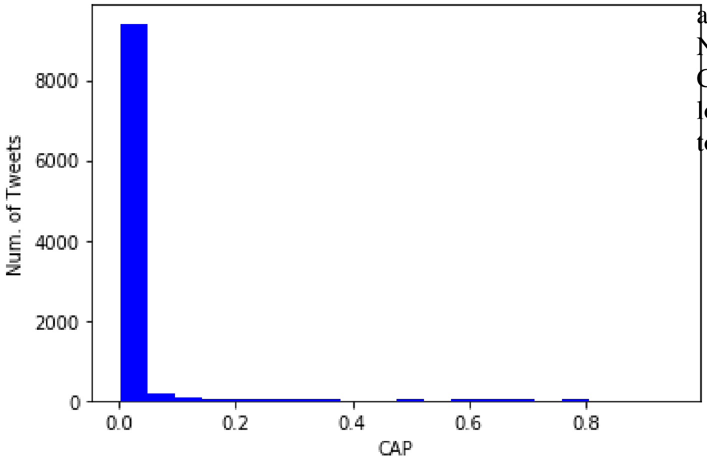

Fig. 2. Histogram of the distribution of Complete Automation Probability (CAP) for tweets in the dataset collected from IUNI's Botometer. CAP values are in the range [0, 1].

presents a new avenue of hate speech detection to explore, and there are several interesting implications. First, the dataset for this task is vital. We used an existing dataset to suit our needs but sampling specifically to produce a dataset of retweet networks could avoid the pitfalls introduced by our data collection methods. Future work should also focus on applications of advanced graph-based machine learning techniques. New graph-based classification techniques as the field continues to grow and may provide models with better performance than our experimental methods. In our experiments we assume that the adversarial attacks come from a single actor producing adversarial attacks against our diffusion-based detection difficult. Future robustness experiments should consider the effects of a coordinated group attacks that could change the sharing patterns.

Examining our methods using other datasets could demonstrate further robustness. For instance, our methods are language agnostic and therefore should find similar results on non-English datasets. Exploring differences in the spread of hate speech contents among different countries and communities could provide for better automated detection systems and behavioral insights into communities. Likewise, we do not analyze the specific accounts associated with hate speech diffusion. Our results show that hate speech tweets tend to be shared rapidly but not propagate broadly, suggesting that perhaps the hateful content remains constrained within an insular community. Investigating community dynamics and communication patterns of hate groups and their followers could provide more features for detection.

Acknowledgment

We would like to first thank Dr. Nir Rosenfeld. We are indebted to Dr. Rosenfeld for his guidance and insights during the research process. We would also like to thank Professor Yaron Singer for his timely advice and perspectives on the research. We further thank Antigoni Founta for her assistance and access to her dataset and finally, the Indiana University Network Science Institute, specifically Matthew Hutchinson, Chathuri PeliKankanamalage, and Clayton Davidson, for allowing us to use their automated bot and network analysis tools.

References

[1] Cox, J., Koebler, J.: Twitter Won't Treat White Supremacy Like ISIS Because It'd Have to Ban Some GOP Politicians Too, https://www.vice.com/en_us/article/a3xgq5/.

[2] Benner, K.: Twitter Adds New Ways to Curb Abuse and Hate Speech, https://www.nytimes.com/2016/11/16/technology/twitter-adds-new-ways-to-curb-abuse-and-hate-speech.html.

[3] Sonnad, N.: Alt-right trolls are using these code words for racial slurs online, https://qz.com/798305.

[4] Ribeiro, M. H., Calais, P. H., Santos, Y. A., Almeida, V. A., Meira Jr, W.: Characterizing and Detecting Hateful Users on Twitter. In: Proceedings of the Twelfth International AAAI Conference on Web and Social Media (2018).

[5] Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated Hate Speech Detection and the Problem of Offensive Language. In: Eleventh International AAAI Conference on Web and Social Media (2017).

[6] Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep Learning for Hate Speech Detection in Tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion (pp. 759-760) (2017).

[7] Gröndahl, T., Pajola, L., Juuti, M., Conti, M., Asokan, N.: All You Need is "Love" Evading Hate Speech Detection. In: Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security (pp. 2-12) (2018).

[8] Waseem, Z., Hovy, D.: Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: Proceedings of the NAACL Student Research Workshop (pp. 88-93) (2016).

[9] Romero, D. M., Meeder, B., Kleinberg, J.: Differences in the Mechanics of Information Diffusion across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter. In: Proceedings of the 20th International Conference on World Wide Web (pp. 695-704) (2011).

[10] Wu, B. and Shen, H.: Analyzing and Predicting News Popularity on Twitter. International Journal of Information Management, 35(6), 702-711 (2015).

[11] Vosoughi, S., Roy, D., Aral, S.: The Spread of True and False News Online. Science, 359(6380), 1146-1151 (2018).

[12] Founta, A. M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., Vakali, A., Sirivianos, M., Kourtellis, N.: Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In: Proceedings of the Twelfth International AAAI Conference on Web and Social Media (2018).

[13] Oak, R.: "Adversarial Examples for Hate Speech Classifiers". Poster presented at the ACM Conference on Computer and Communications Security in London, United Kingdom (2019).

[14] Goel, S., Watts, D. J., Goldstein, D. G.: The Structure of Online Diffusion Networks. In: Proceedings of the 13th ACM Conference on Electronic Commerce (2012).

[15] Yanardag, P., Vishwanathan, S. V. N.: Deep Graph Kernels. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1365-1374) (2015).

[16] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Detecting Hate Speech in Social Media. In: Proceedings of Recent Advances in Natural Language Processing (2017).

[17] Davis, C. A., Ciampaglia, G. L., Aiello, L. M., Chung, K., Conover, M. D., Ferrara, E., Flammini, A., Fox, G. C., Gao, X., Gonçalves, B., Grabowicz, P. A., Hong, K., Hui, P., McCaulay, S., McKelvey, K., Meiss, M. R., Patil, S., Peli Kankanamalage, C., Pentchev, V., Qiu, J., Ratkiewicz, J., Rudnick, A., Serrette, B., Shiralkar, P., Varol, O., Weng, L., Wu, T., Younge, A. J., Menczer, F. OSoMe: the IUNI Observatory on Social Media. PeerJ Computer Science, 2,e87, 2376-5992 (2016). https://doi.org/10.7717/peerj-cs.87.