Harnessing the Nature of Spam in Scalable Online Social Spam Detection

Hailu Xu, Boyuan Guan, Pinchao Liu, William Escudero, Liting Hu
School of Computing & Information Science
Florida International University
Miami, Florida, USA
Email:{hxu017, bguan003, pliu002, wescu001, lhu}@cs.fiu.edu

***Abstract*—Disinformation in social networks has been a worldwide problem. Social users are surrounded by a huge volume of malicious links, biased comments, fake reviews, or fraudulent advertisements, etc. Traditional spam detection approaches propose a variety of statistical feature-based models to filter out social spam from a historical dataset. However, they omit the real word situation of social data, that is, social spam is fast changing with new topics or events. Therefore, traditional approaches cannot effectively achieve online detection of the ”drifting” social spam with a fixed statistic feature set. In this paper, we present Sifter, a system which can detect online social spam in a scalable manner without the labor-intensive feature engineering. The Sifter system is two-fold: (1) a decentralized DHT-based overlay deployment for harnessing the group characteristics of social spam activities within a specific topic/event; (2) a social spam processing with the support of Recurrent Neural Network (RNN) to get rid of the traditional manual feature engineering. Results show that Sifter achieves graceful spam detection performances with the minimal size of data and good balance in group management.**

***Keywords*-online social networks; spam detection; RNN;**

I. Introduction

With billions of people active in social communities, social networks have become the main source of news and public events. According to a report, 62% US adults currently acquire news and information from social networks [2]. The proliferation of social network is built on shared activities and comments by public users. However, it has become common place for spreading fakes news, fraudulent advertising, propagating political rumors, biasing product values, and even inducing democratic chaos. Examples like fake news in Facebook [1], cheating reviews in Yelp [10], ISIS propaganda distribution [15], and Charlottesville Chaos [16] demonstrated the serious consequences of biased and actively spam. Therefore, social spam detection has been a key issue in current online social networks.

To achieve efficient spam detection, the characteristics of current social spam should be well analyzed. One characteristic of social spam is that malicious activities are mostly concentrated in a small number of groups. This is because spam posts are normally published by a number of spammers and their employees, which reflects group behaviors. For example, research shows that 17 groups account for 75% of the spam among 5 million posts in Twitter [4]. Another research also points out that spam activities are manipulated by various sparse or dense group [23].

Another characteristic of social spam is: spammers' activities normally target on the most current events and typically utilize these events to enlarge their influences. For example, a hacked official Twitter account of the Associated Press claimed that two new explosions in the White House took place and the President was injured, just after the Boston explosion [27]. Under the shadow of explosion, this post was immediately trusted by the public and caused a serious public panic. The close connection between spam activities and social events has made the spam posts highly deceptive and greatly increased the difficulty of spam detection [11, 14].

Traditional offline social spam detection typically uses learning algorithms incorporating with a static feature set. Inevitable, this approach faces several limitations when dealing with online social data. First, the static feature set from a specific data source is difficult to be extended to deal with the data from new sources (e.g., new topics or different platforms). For example, research shows that the user features can work well in dealing with Social Honeypot Dataset, with F1 score closes to 94%. However, for the 1KS-10KN dataset, user features can only get the F1 score near 79%. Similarly, another kind of features such as the N-gram features in the 1KS-10KN dataset can achieve an F1 score of over 82%, but in Social Honeypot Dataset, the F1 score can only reach 70% [20]. Furthermore, many prior studies focused on extracting complicated features, e.g., finding anomalous patterns of pronouns, conjunctions, emotional words [26], user credibility [19], etc. However, these features are too specific and not strongly discriminant among drifting online social data [22].

Second, online spam detection requires relatively short delays, however, traditional spam detection may take days or even months to complete. The delay is mainly from the feature engineering [18], where researchers need to spend a lot of time to extract and verify the features. Though some studies explore the online/real-time spam detection [5, 8, 17], they either specifically focus on URL detection (which limits their use in the real-world situation [8, 17]) or they cannot get rid of the labor-intensive feature engineering [5] (which incurs long latency and is not feasible for the online spam detection). Besides, most successful applications of online processing in social media only focus on topic or event recognition [9]. However, these methods do not show the prospects in detecting

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

event spam and spam manipulators.

In this paper, we introduce Sifter, a system to unmask online social spam by utilizing the group characteristic of social spam within a specific topic/event. Sifter implements Recurrent Neural Network in the data processing which can process the time sequential social data streams to achieve automatic data processing. Besides, Sifter clusters social data related to one topic/event into an application-level group. By processing the data for a topic/event in one specific group, Sifter can achieve effective spam detection in a scalable way with minimal data processing.

The remainder of this paper is organized as follows. Section II discusses the background of RNN. Section III describes Sifter's design. Section IV evaluates Sifter with experiments. Related work is presented in Section V. We finally conclude with directions for future work in Section VI.

II. Recurrent Neural Network and LSTM

In this section, we first present the details of Recurrent Neural Network and the long-short term memory (LSTM), then introduce the motivation of implementing RNN in social spam detection.

Recurrent Neural Network (RNN) is a rapidly emerging architecture originally from the traditional artificial neural network (ANN) [7]. The characteristic that differentiates RNN from other neural networks is the connections between nodes in the hidden layer form a directed graph along a sequence. Numerous applications have shown RNN is very good at predicting the next character in the text or the next word in a sequence, and can be used for complex tasks [7]. For instance, traditional ANNs can solve the "filling-the-blank" problem, e.g., give a solution of which word should be embedded into "Tom leaves Washington, he is now in _", but the solution may contain "Washington" since via the N-gram (n words before the blank, n is typically 3 or 4) only contains "he is now in" and cannot have the memory of former sentence. In contrast, RNN can utilize the memory of historical words (e.g., "leave Washington"), gives a correct prediction which should not have "Washington" in the blank of the sentence.

RNN can only have a short memory of few terms, to overcome this limitation, a new long short-term memory (LSTM) networks is presented, which uses special hidden units, the natural behavior of which is to remember inputs for a long time [7, 6] is presented. LSTM uses gates (i.e. forget gate, input gate, and output gate) to control which part of former memory results can be utilized in the current computation, and then decide which part of output can send to the next computation.

In this study, inspiring by the success of RNN in dealing with the time sequential applications, e.g. speech recognition [12], fake news detection [11], etc., we propose to use RNN to process the social spam posts. RNN is well-fitted to the social spam detection for two reasons: first, social network data is based on time sequences [13], i.e. posts are sequential in nature. Second, the training data is of variable length, i.e. the number of posts will vary in different time slots. RNN can conveniently handle the length variable input with the neural network architecture. Besides, the characteristics of neural networks avoid the labor-intensive feature engineering since neural network can automatically extract proper features in the training process.

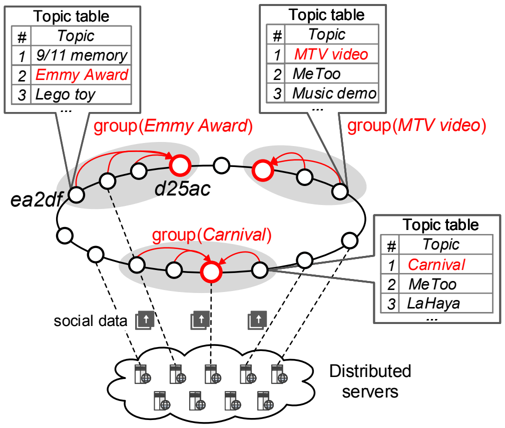


Figure 1. Topic/event-based group management in Sifter.

III. Sifter Design

The Sifter system consists of three major components: (1) the Sifter node; (2) the sifter group; and (3) the Sifter spam detection unit (SDU).

The first component is the Sifter Node. Each Sifter node is assigned a unique, 128-bit *nodeId* in a circular *nodeId* space ranging from 0 ∼ 2^{128}-1. All nodes' *nodeIds* are uniformly distributed. Given a message and a key, the message can be guaranteed to be routed to the node with the *nodeId* numerically closest to that key, within $\lceil log_{2^b} N \rceil$ steps, where b is a base with a normal value 4. Besides, each node maintains a topic table, which is used to store the topics or events collected from the local social media data sources. For example, if one node gathers social data mainly from five topics '*MeToo*', '*LaHaya*', '*EXO*', '*FelizLunes*', and '*Emmy Award*', it will save these five topics instances into its topic table. Sifter group is then created via the topic table and works with the topic-based spam identification in the continuing processing. The second component is the Sifter Group. The Sifter group mainly responses for the whole spam detection task of one specific topic/event. At the beginning, Sifter allows node to create a topic/event-based group via its topic table. The Sifter group management is fulfilled by Scribe methodology which is an application-level group communication system built upon DHT-based overlay [3]. Sifter uses a pseudorandom Pastry key to name a group, called *groupId*. Usually, the *groupId* is the hash of the topic/event's textual name concatenated with its creator's name. Sifter defines that only the node who collects the data from the specific topic can join the topic-based group. As the Fig. 1 shows, for example, node *ea2df* identifies that it receives the social data from the topic '*Emmy Award*'. It then automatically routes a JOIN message towards the group which has the *groupId* '*Emmy Award*'. The message will continue to be routed until it reaches the node *d25ac* in that group. The route traversed by the message to the group would be added. As a result, Sifter can efficiently support large numbers groups, arbitrary numbers of group members, and group with highly dynamic membership.

In Sifter, multiple nodes join in one group and maintain a functional tree. The root node of a group tree responses for the main control flow of the whole group. The group root orchestrates the parent nodes and leaf nodes by multicasting different commands to them. For example, as shown in Fig. 2 left, when the root node starts to aggregates results, it requires the following nodes to deliver their results to the upper layer. Then the group functional tree progressively rolls up results until reach the root.

Besides, the group root is responsible for the consistency of RNN models. The group root will communicate with all other group members to acquire different features from RNN models. Then the group root will adjust the weights and biases by evaluating all features. Finally, the group root multicasts the optimal model features (e.g., weights and biases) to all other members and lets them update their models. With the consistent models, each group can effectively handle the social data within a specific event.

The third component is the Sifter spam detection unit (SDU). As shown in Fig. 2 right, each leaf node can maintain its own Recurrent Neural Network, acts as a SDU. A SDU is used for the processing of social data and filtering out the spam posts. The leaf nodes can collect the social data from various local data servers through social network APIs. Each leaf node promptly updates its topic table based on the current popular topics from the local data server. Since one event typically lasts for a few days, the topic table is not very frequently updated. Besides, Sifter sets a threshold for the topic table to prevent the nodes from joining or leaving one group too frequently. Each SDU maintains its own Recurrent Neural Network architecture, which can efficiently process the time sequential social data.

IV. Evaluation Results

Sifter assembles each leaf node with a separate RNN architecture for social data processing. For achieving good performances and detecting social spam with long series of logs, Sifter implements LSTM in leaf node. Considering that treating each time-stamp as an input to a cell should be extremely inefficient and reducing utility [14], we propose to partition the data into segments, each of which will be an input to a cell. We apply a natural partitioning by changing the temporal granularity from different time intervals. In each interval, we use the $tf * idf$ values of the vocabulary terms as input. We prune the vocabulary by keeping the top-K terms according their $tf * idf$ value, so the input dimension is K.

Sifter is evaluated on a testbed of 800 agents hosted by 20 servers running on Linux. Each server has a QEMU Virtual CPU with 3.4GHz processor, 4G of memory and 30 GB hard drives. The system was implemented in Java by using Java SE Development Kit 7 in x64, version 1.7. We initially evaluate the Sifter system with scaling with 1GB posts from Twitter in 800 agents and evaluate the spam detection performance in a three-layer LSTM model with a sample dataset (data was collected from Twitter in 2017) which consists of 50,000 posts (37465 posts are Ham and 12535 posts are Spam) [25].

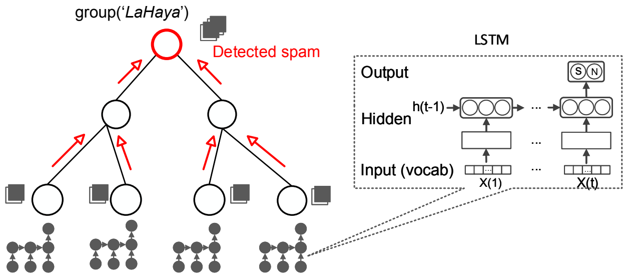


Figure 2. The Sifter group functional tree and RNN in SDU.

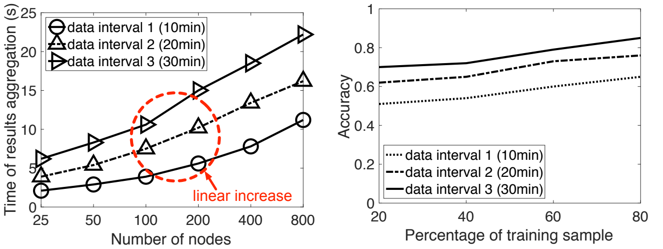
The Sifter group is responsible for the processing of topic/event-based social data and the group functional tree organizes the whole detection processing. Therefore, group management is a key component during the entire process. We first evaluate the results aggregation time in Sifter group. We process the original data with different time interval (i.e. 10, 20, 30 min). The results are shown in Fig. 3a. As the figure shows, when Sifter uses the same dataset but with a different number of nodes (i.e. 25, 50, 100, 200, 400, 800), the time of results aggregation linearly increases, rather than fold increases. This is because the linear increment of the delivery or reception time is strictly determined by the tree depth $O(logN)$, which further reflects that the group functional tree in the overall scalable processing exhibits a very good balance.

Table I shows the detection performance in Sifter. Results show that Sifter can achieve good performances in detecting social spam from online time sequential data. Besides, Fig. 3b represents the detection accuracy with different size of training data. From the figure, we can see that with the increased size of training sample, the overall detection accuracy increases. Moreover, since the Recurrent Neural Network completes the detection without labor-intensive feature engineering, it presents graceful prospects in dealing with the time sequential social data.

TABLE I. Results of spam classification.

Data blocks	F1	Precision	Recall
data interval 1 (10min)	82.0%	0.903	0.751
data interval 2 (20min)	84.7%	0.912	0.791
data interval 3 (30min)	89.6%	0.923	0.87

In the future, we will thoroughly evaluate the Sifter system from multiple aspects. For instance, we will evaluate the data processing latency, group fault tolerance, system load-balance, etc. Besides, we will explore the relationship between detection performances and the RNN model architectures. And we will achieve efficient spam detection with a distributed Recurrent Neural network in the Sifter system.



(a) Time of root aggregates results. (b) Accuracy vs. percentage of training sample

Figure 3. Results in Sifter. (a) shows the results aggregation time in Sifter group. (b) shows the accuracy with different size of training data.

V. Related Work

Many prior studies focus on offline social spam detection. They typically mine social spam from an offline dataset via the content features [20, 24], user behavior [20, 26], or social connections [5], etc. Different from them, we implement Sifter via online social data processing, which can catch up with the most current social spam activities. Besides, several studies implement online social data processing. For instance, they identify online social trends [27] or filter out online social spam by URL information [8, 17, 22]. Different from them, we implement online social spam detection from a broader scope of social data, which could be more useful in the real world.

Several recent studies analyze the social data by cooperating with neural networks to achieve efficient processing. For instance, Ma et al. identify social rumors by utilizing RNN with the time sequential social data [11]. CSI [14] detects social rumors by integrating the group behavior and article engagement. EANN [21] explores the role of generalized event features in the social rumor detection. Inspired by the effectiveness of RNN in time sequential social data, we implement RNN in social spam detection without incurring labor-intensive feature engineering, and further explore the spam detection in a scalable way.

VI. Conclusion

In this work, we preliminarily present an online spam detection system (Sifter), a distributed and scalable system that can detect social spam in an online fashion. Implementing with the Recurrent Neural Networks, Sifter can effectively harvest and discover the general characteristics of social spam without the labor-intensive feature engineering. Besides, Sifter aggregates various social data into different groups based on their related topics/events. By topic-based group management, Sifter can efficiently filter out social spam from one topic with minimal data processing.

Future work on Sifter will lead to more detailed implementations. We will explore the entire processing latency and balance the scale and the latency of distributed agents in the system. Besides, we seek to reduce the runtime overhead, achieve load-balance with highly efficient data processing, and support social spam detection across various platforms, etc.

References

[1] Hunt Allcott and Matthew Gentzkow. “Social media and fake news in the 2016 election”. In: *Journal of Economic Perspectives* (2017).

[2] Alessandro Bessi and Emilio Ferrara. “Social bots distort the 2016 US Presidential election online discussion”. In: (2016).

[3] Miguel Castro et al. “SCRIBE: A large-scale and decentralized application-level multicast infrastructure”. In: *IEEE Journal on Selected Areas in communications* (2002).

[4] Chao Chen et al. “Investigating the deceptive information in Twitter spam”. In: *Future Generation Computer Systems* (2017).

[5] Chao Chen et al. “Statistical features-based real-time detection of drifted twitter spam”. In: *IEEE Transactions on Information Forensics and Security* (2017).

[6] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to forget: Continual prediction with LSTM”. In: (1999).

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* (2015).

[8] Sangho Lee and Jong Kim. “Warningbird: A near real-time detection system for suspicious urls in twitter stream”. In: *IEEE transactions on dependable and secure computing* (2013).

[9] Xiaomo Liu et al. “Reuters tracer: A large scale system of detecting & verifying real-time news events from twitter”. In: *In the 25th ACM CIKM*. 2016.

[10] Michael Luca and Georgios Zervas. “Fake it till you make it: Reputation, competition, and Yelp review fraud”. In: *Management Science* (2016).

[11] Jing Ma et al. “Detecting Rumors from Microblogs with Recurrent Neural Networks.” In: *IJCAI*. 2016.

[12] Yajie Miao, Mohammad Gowayyed, and Florian Metze. “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding”. In: *2015 IEEE Workshop on ASRU*.

[13] Bhavtosh Rath et al. “From retweet to believability: Utilizing trust to identify rumor spreaders on Twitter”. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM. 2017.

[14] Natali Ruchansky, Sungyong Seo, and Yan Liu. “Csi: A hybrid deep model for fake news detection”. In: *2017 ACM CIKM*.

[15] Scott Shane and Ben Hubbard. “ISIS displaying a deft command of varied media”. In: *New York Times* 30 (2014).

[16] “So Much Trump Chaos.” https://www.nytimes.com/2018/03/02/opinion/trump-gun-control-nra.html.

[17] Kurt Thomas et al. “Design and evaluation of a real-time url spam filtering service”. In: *2011 IEEE Symposium on Security and Privacy (SP)*.

[18] “Understanding Feature Engineering -Continuous Numeric Data”. https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b.

[19] Marco Viviani and Gabriella Pasi. “Credibility in social media: opinions, news, and health information-a survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2017).

[20] Bo Wang et al. “Making the most of tweet-inherent features for social spam detection on Twitter”. In: *arXiv preprint arXiv:1503.07405* (2015).

[21] Yaqing Wang et al. “EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection”. In: *In the 24th ACM SIGKDD*. 2018.

[22] Mahdi Washha et al. “A Topic-Based Hidden Markov Model for Real-Time Spam Tweets Filtering”. In: *Procedia Computer Science* (2017).

[23] Liang Wu et al. “Adaptive Spammer Detection with Sparse Group Modeling.” In: *ICWSM*. 2017.

[24] Hailu Xu, Weiqing Sun, and Ahmad Javaid. “Efficient spam detection across online social networks”. In: *Big Data Analysis (ICBDA), 2016 IEEE International Conference on*.

[25] Hailu Xu et al. “Oases: An Online Scalable Spam Detection System for Social Networks”. In: *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*.

[26] Qiang Zhang et al. “Spam comments detection with self-extensible dictionary and text-based features”. In: *Computers and Communications (ISCC), 2017 IEEE Symposium on*.

[27] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. “Enquiring minds: Early detection of rumors in social media from enquiry posts”. In: *Proceedings of the 24th International Conference on World Wide Web*. 2015.