Detecting spam campaign in twitter with semantic similarity

M Mostafa¹, A Abdelwahab¹², H M Sayed¹
¹Department of Information System, Faculty of Computer and Information, Menoufia University, Menoufia, Egypt
²Department of Information Systems, College of Computer science and Information Technology, King Faisal University, Saudi Arabia

Abstract—Twitter is a widespread supply for real-time news distribution between individuals. Furthermore, spammers could post any kinds of spam content to users, and a variant of incidents are committed on Twitter against users. These threats aren’t restricted to the social media platforms however they threaten the safety of Twitter users. Most of the researches used deep learning techniques to detect Twitter spammer activities. The traditional solutions check the behavior of each account or campaign of similar purpose accounts. The number of solutions concentrate on detecting spam campaign based on URL only and ignoring text in a tweet. In this paper, the lightweight framework is proposed to take tweet text into consideration for optimizing spam campaign detection methods based on deep learning techniques. The main contribution of this work summarized in two key points. First one is to summarize text of the tweets to cluster them. The second one is to find similar tweets based on Siamese Recurrent Network. Experimental results show the ability of the proposed technique to extract the right candidate campaign and classify them as spam or not with high recall and precision.

1. Introduction
In recent years, millions of internet users can communicate and collaborate within Online Social Networks (OSNs). Twitter is considered the most popular social network which provides a free blogging services for users to spread their news and ideas within 280 characters. Users can follow others and so on, through different platform [1]. Every minute, thousands of Twitter users share their status and news about of their discoveries [2]. Furthermore, Twitter platform also attracts criminal accounts (spammers) that can post spam contents which may include harmful URLs. These URLs could redirect users to malicious or phishing websites for making money illegitimately [3, 4] by attacking user’s platform. As Tweeter put limits on the length of Tweets, it makes spammer to cheat users by putting cheat text or malicious URL to redirect to the external website [5]. In a study of comparison between the email spam and social spam, the click-through rate of Twitter spam reaches 0.13%, although in email spam reaches 0.0003%–0.0006% [6]. Furthermore, the social spam is considered more dangerous and cheat a lot of users [7].

In order to maintain this problem, several studies of spam detection are focusing on the message or account level. However, these detection approaches always check each Tweet contents or URLs to classify whether it contains spam or not. But these approaches could not generate a comparable result since they depend only on one algorithm was used in its mechanism. Recently, a lot of researches
concentrate on building binary classifiers with the input of statical features [3, 7]. The features could be generated from Twitter’s Streaming APIs and picked by a JSON object, and they include user-level attributes (such as digits, quantity of URLs, hashtags in the tweet respectively) and account-level attributes (like age of account, number of followings and amount of followers)) [8,9,10]. Simultaneously, blacklisting techniques are extremely time-consuming due to individual’s participation for unsolicited information recognition. Consequently, these challenges contribute to the motivation of our work.

To deal with those problems including single supporting algorithm, feature extraction issue, accuracy shortage and low speed, an effective classification method based on deep learning is proposed by this paper. Firstly, we apply Sen2Vec to pre-process the tweets instead of feature extraction, where the technique adopted is an advanced language processing method in deep learning and it can convert word or document to representative vector [11]. Afterward, a machine learning model is built on the basis of several machine learning algorithms to distinguish spam and non-spam. At the next stage, parameter setting is assigned for spam filtering. The experiments are set up with a real-world ground-truth dataset. The following step refers to compare our classification outcome to those also analyse the content of tweets. Finally, we make a further comparison between the new method and current detection technique which do not rely on text analysis with respect to accuracy. As a result, our innovative methods are proved to outperform them.

Consequently, these challenges inspire us to go through in this work to contribute in campaign detection. The main contribution of this paper is summarized as follows, framework is maintained to detect spam campaigns based in deep learning using semantic similarity of text in Tweets. Therefore, to link all similar accounts that posted same content, significant steps are followed.

- Represent each Tweet into sentence embedding (sent2vec) [11].
- Define the relevant Tweets and apply text similarity for these Tweets and Then, build a graph with all similar purpose content [12].
- Then, apply cohesive campaign extraction to extract campaign.
- Classify them for spam or ham campaign.

The remaining of this paper is composed as follows. Section 2 starts with brief in literature review on Twitter spam detection. section 3 will explain in details our innovative semantic method. In Section 4, experimental settings and performance results are shown. In Section 5, some points will be discussed which can affect the classifier performance. Finally, References are represented in section 6.

2. Related work
A lot of researches has been applied to standing in front of the problem of social spamming. This work is categorized to three categories as defined in figure(1), based on survey [13] of approaches and challenges which categorize them into three main categories syntax analysis, feature analysis and blacklist techniques.
2.1 Detection based on Syntax Analysis

These methods focuses on analyzing Tweets at word or character based. A lot of researchers focusing on checking shortened URLs in each Tweet. Most of the spammers use shortened URLs to hide their spam URLs, that they generate it form ordinal shortened service to authorize it among users. Lee and Kim [14] have proposed a novel technique for detecting Twitter malicious URL based on information ratio for suspicious URLs, but it stuck from dynamic redirection. Wang, et al. [15] build a dataset of click-through rate feature to classify whatever shortened URLs is spam or not spam. However, some approaches focus on tweet content. for instance, Tang, et al. [16] apply the text in tweet content for a deep learning model to build embedded word vector from its context to classify them as spam or not. However, Tang, et al.’s method has a good performance measure with (F1-measure 87.60%<90%), although, it still has area for optimizations. Rybina [17] also prove that learning linguistic analysis based on document-level modelling is better than word level, to build text classification. Although, the text classification model didn’t have various machine learning models, which make it difficult to check various performance results. although, this paper [18] build a text classifier using ensemble learning approach which contains different machine learning methods. Therefore, these two paper suggested that the most efficient classifier is Na¨ıve Bayes because it give a good result in terms of accuracy and time cost (F1-measure >= 90%)

2.2 Detection based on feature analysis

Feature Analysis based functions depend on building statistic features from account or/and tweet to be used as training input features for different classifiers in machine learning as defined in these works [19, 20, 21, 22, 23,24]. Account features are defined as the user account age, the number of followers, and the followings number. Tweet features contain average number of words in each tweet, the average number of hashtags in a Tweet, the ratio of account Tweets which include URLs and etc. Those techniques for feature extraction were build and used as shown. For instance, Benevenuto, et al. [19] use $\chi^2$ method to analyse the most ten important attributes, and Chen, et al. [20] use Twitter’s JSON object to collect the important features. After building the features dataset, it is important to choose the best classification algorithm to be used for training the extracted features. Therefore, a lot of different Twitter spam detection technique focus on determining the optimal machine learning methods. For instance, Wang[21] choose Naive Bayesian algorithm to detect spammers which produce high precision of 89%. Benevenuto, et al. [22] employed Kernal Support Vector Machine algorithm to detect spammers based on real-world datasets. Stringhini, et al. [23] apply the Random Forest process to build its spam classifier. Lee, et al. [25] use honeypots to obtain spam profile feature, which trained them with various machine learning algorithms, for instance LogitBoost and Decorate [14]. The feature-based machine learning classifiers have two major problems. The first one is spam drift which is critical problem, that can affect the precision of classifier prediction [22, 26]. Liu, et al. [24] solved this problem later by proposing a new method which combined two different methods fuzzy-based redistribution to generate more spam samples with asymmetric sampling to balance the size of spam and not spam features in dataset. The second one is the problem of data collection processes, that it suffers from feature fabrication. To address this problem, many researchers are using advantages of social graph to detect robust features to help them in avoiding feature fabrication as proposed in [17, 18,23]. Jin, et al. [23]
define social network features as two category characteristics of user account and structure behaviour of graph. Song, et al. [17] build Twitter social graphs and use the distance between pair of social users to expose if spam. Yang, et al. [18] generated social graph based on local clustering coefficient, bidirectional links ratio and betweenness centrality. With the usage of social graphs, social graph statistical features can be analysed to detect spam account. Therefore, this generated features are examined to be one of most efficient technique which achieved precision of 95.1%. However, this method can’t be used in real world because it is impractical to collect all data about each account in Twitter which has millions of users.

2.3 Blacklist Techniques
Blacklist techniques depend on web filter approaches to detect spam behaviour, according to some information analysis on websites crawling or user feedback, they build the classifier for blocking malicious websites. Ma, et al. [22] presented a novel solution which is better than same-purpose classifiers. Oliver, et al. [16] recognized harmful URLs by using blacklisting approach which depends on a third party. Although, this method is efficient in the detection of spammers but it depends on manual labelling making it very time consuming method.

In a nutshell, the existing spam detection techniques on Twitter are still not able to recognize spamming activities accurately and quickly with the metrics of Precision, Recall, and F1-measure. So to achieve a novel results with better performance and get less time consuming prompt us to the motivation of this work.

3. Campaign detection framework
This section will describe our new campaign detection model based on vector-based characteristics to train sentence embedding. Our model is inspired from [10], which depend on three main steps. The first step is to calculate the similarity of all accounts that post tweets on similar purposes to be used in constructing the graph. The second step is to extract cohesive graphs as candidate campaign. The third step is to classify this campaign as legitimate or spam campaign.

![Figure 2. Our proposed detection framework with adding semantic similarity modification](image)

The problem with the first step, that it builds the model based on statistics, that it defines the similarity between accounts using Shannon theory, which assumes that the more users posting similar purpose tweet with low probability, the more information they share. Therefore, account similarity model can be modified with the new model by using semantic account similarity as shown in figure (2) (the modification is highlighted with a rectangle). Which give us good results than the exist model.
3.1 Tweet representation

Tweeter has at least 6000 tweets posted, by each second [8]. So, our model uses semantic similarity to compare all pair of tweets to build the graph. Unsupervised model is used to represent Tweets [11], called sent2vec. It is a simple technique but efficient to train distributed representation of sentence, that it defines sentence embedding as the average of source word [27] to its neighbor words as in (1). That it learns source embedding from n-gram of each word in sentence and get average the n-gram embedding along with the target words as in (2). Then it applies softmax output function followed by negative sampling as in (3). Which help us to improve training efficiency. That it not changing the weights each time because we have to predict huge number of output classes, coupling binary logistic function with negative sampling, the unsupervised training objective is formulated as follows:

$$
\min_{U,V} \sum_{S \in C} F_S (UV_{LS})
$$

(1)

$$
v_s = \frac{1}{|R(s)|} V_{LRS} = \frac{1}{|R(s)|} \sum_{w \in R(S)} V_w
$$

(2)

$$
\min_{U,V} \sum_{S \in C} \sum_{w_t \in S} (l(U^T_{w_t} V_{S \setminus (w_t)}) + \sum_{w' \in N_{w_t}} l(-U^T_{w_t} V_{S \setminus (w_t)}))
$$

(3)

where $S$ corresponds to the current sentence and $N_{wt}$ is the set of words sampled negatively for the word $w_t \in S$. this model has a lot of advantages that it has low computational cost for training which make it powerful in representing this huge number of Tweets that posted daily. After the representation of the Tweets, cosine distance between each pair of Tweets is calculated to indicate if they exceed the similarity threshold $T_s$, the nearest pairs are entered to the next layer to calculate semantic similarity as will be described next.

3.2 Tweets similarity

In this step, Manhattan lstm model is used to get the similarity between two Tweets [12], that it depends on two lstm network with solely tied weight as in Siamese architecture $L_{stma} = L_{stmb}$, that each network learns embedding from sequence of word vector with $(din=300, drep=50$ in this work). $X_1, \ldots, x_T$ are passed to the network which updates the hidden layer, then Manhattan distance is calculated between each sentence representation $h_{Ta(a)},h_{Tb(b)}$, then the output of them passed to sigmoid activation function to extract the similarity of them. Next, the graph $G=(V,E)$ in which $v$ is the vertex denoted by the account of the Tweet and $E$ the edge built if the similarity $S_{a,b} > S_{min}$. Accordingly, we apply cohesive extraction approach to extract subgraph $G'$, that it searches for the max number of edges with the same number of vertices of $G'$.

Next, the candidate campaign will be used as input feature to several machine learning classifiers, such as SVM or naïve bayes or random forest. In order to predict the candidate campaign as a normal or spam campaign [9].

![Figure 3. This is Manhattan LSTM Model which Read each Word-Vector from each tweet pair, that the similarity between these two representation is used to predict semantic similarity](image)
4. Analysing the results of detection framework

In this section, the performance results of our modified framework will be showed and discussed. That we will show the results of comparison between the two models and show the result of comparison between each classifier.

4.1 Ground-Truth twitter dataset

To evaluate our proposed spam campaign detection. A real-life 3-day from twitter is applied to obtain the ground-truth, which contain 384,753 Tweets after filtering this Tweets with java program, Tweets less than 75 chars and hasn’t contained valid URL are ignores as it can’t provide any meaning. We got 150,000 Tweet, then pre-processing phase is applied for all Tweets to ignore all special characters and emotions, and replacing Tweets shorten URL to the origin to be able to compare this links to text similarity phase with using cross-validation 10 fold.

4.2 Basic parameter setup

To perform the whole framework results, we run our experiments on Linux Ubuntu 18.04 operating system with Intel(R) Core(TM) i7-4702MQ CPU @ 2.20 GHz of 12 GB memory. At the first step twitter representation we use sent2vec model, which will be assigned with pre-trained English Tweets as unigram with 700-dimension vector.

4.3 Evaluation metrics

To evaluate our detection framework, variation of average recall, average precision, average F1-measure are measured [10] to notice the accuracy of the extracted campaign. The average precision is defined as:

\[
\text{APrecision} = \frac{1}{n} \sum_{i=1}^{n} \max_{j \in 1,2,...,m} \frac{|C_i \cap T_j|}{|C_i|}
\]

(4)

Where \( C_i \cap T_j \) is the number of correctly predicted campaigns, \( n \) is the number of extracted campaign and \( m \) the number of the real campaign in the dataset \( C_i \) and \( T_j \) for an actual (real) campaign. For each extracted campaign, its precision is determined by the actual one which has the largest common accounts. Similarly, the (ARecall) average recall is defined as:

\[
\text{ARecall} = \frac{1}{m} \sum_{i=1}^{m} \max_{j \in 1,2,...,m} \frac{|C_i \cap T_j|}{|T_j|}
\]

(5)

where the recall for each real campaign is measured relative to the extracted campaigns which contains the most same accounts. Furthermore, the average F1 measure (AF) is measured by ARecall and APrecision as:

\[
AF = \frac{2 \times \text{APrecision} \times \text{ARecall}}{\text{APrecision} + \text{ARecall}}
\]

(6)

For classifying these extracted campaign, non-linear SVM algorithm is applied with radial basis function (RBF) kernel to determine spam users.
4.4 Effectiveness of semantic similarity
Tweets dataset are applied on the two-step semantic similarity function. We found that the similarity in the first step (using sent2vec model) is very high if the distance between the Tweets less than $S_{ij} < 0.087$ as in figure (4), which we represent all Tweets in our dataset using sent2vec model. So that to plot this figure principle component analysis (PCA) is used to project all these Tweets into two components to able to visualize in 2d plot. After that, the list of pairs results is entered to the next step (Manhattan lstm model) to recalculate the similarity of this pair. Besides, a graph of accounts with all similar purpose of Tweets pair are extracted to 1948 candidate campaign. As shown in figure (5). According to this step, all Tweets with less than 13 accounts are eliminated as they can’t be effective campaign to classify it as shown in figure (4). To achieve the results, small dataset is selected to manually get labelled campaign, so random 930 accounts is choosed and then, compare each pair of accounts to label then if they are posting in similar purpose or not, accordingly, we found 6278 pair of account sharing similar purpose whatever they share same text or URL. We found 58 candidate campaign with results as in table (1) as we compare them with manually labelled.

| # campaigns | APrecision | ARecall | AF  |
|-------------|------------|---------|-----|
| 58          | 0.945      | 0.93    | 0.946 |

4.5 Effectiveness of campaign classification
From the results of the small dataset, we can assume that all 1958 extracted campaign from the all real-life dataset is the actual one with no missing. After that, a SVM classifier is applied and get 1727 are legitimate campaign, 231 are spam campaign. We show the comparison between the original framework and after adding our modification, the performance result are optimized as shown in the results of precision and recall after adding our modification is greater than the original framework.

|                       | APrecision | ARecall | AF  |
|-----------------------|------------|---------|-----|
| U&T Based Model       | 0.909      | 0.873   | 0.89 |
| Ours Model            | 0.929      | 0.89    | 0.91 |

Table 1: Effectiveness of Campaign Extraction Method.

Table 2: Comparison Result of Different Campaign Extraction.
5. Conclusion
In this paper, we studied the issues of twitter spam detection models and propose an innovative method to maintain a framework to improve the similarity estimation method with adding semantic similarity information, by representing all the Tweets with the model that can work fast and effective, that can be able to get similar purpose content and then get each pair to apply sentence similarity to ensure that this pair are similar. then we build the graph of all similar accounts to extract the campaign to classify them to spam or normal account. Therefore, this modification improved the result with high accuracy as we shown in the last section.

As future work, we plan to add the Tweet timestamp to get the similar content at time, we need to improve our text similarity method to be able to extract similarity also with new strange words in Tweets and we need to apply more datasets with less bias to get rigorous results also, we need to try to solve spam drift problem by adding ensemble learning to our detection framework.

6. References

[1] Chu, Z., Widjaja, I. and Wang, H., 2012, June. Detecting social spam campaigns on twitter. In International Conference on Applied Cryptography and Network Security (pp. 455-472). Springer, Berlin, Heidelberg.
[2] Ghosh, S., Viswanath, B., Kooti, F., Sharma, N.K., Kormal, G., Benevenuto, F., Ganguly, N. and Gummadi, K.P., 2012, April. Understanding and combating link farming in the twitter social network. In Proceedings of the 21st international conference on World Wide Web (pp. 61-70). ACM.
[3] Zhu, Y., Wang, X., Zhong, E., Liu, N.N., Li, H. and Yang, Q., 2012, July. Discovering spammers in social networks. In Twenty-Sixth AAAI Conference on Artificial Intelligence.
[4] Adewole, K.S., Anuar, N.B., Kamsin, A., Varathan, K.D. and Razak, S.A., 2017. Malicious accounts: dark of the social networks. Journal of Network and Computer Applications, 79, pp.41-67.
[5] Lee, S. and Kim, J., 2013. Warningbird: A near real-time detection system for suspicious urls in twitter stream. IEEE transactions on dependable and secure computing, 10(3), pp.183-195.
[6] Grier, C., Thomas, K., Paxson, V. and Zhang, M., 2010, October. @ spam: the underground on 140 characters or less. In Proceedings of the 17th ACM conference on Computer and communications security (pp. 27-37). ACM.
[7] Thomas, K., Grier, C., Ma, J., Paxson, V. and Song, D., 2011, May. Design and evaluation of a real-time url spam filtering service. In 2011 IEEE symposium on security and privacy (pp. 447-462). IEEE.
[8] Krikorian, R., 2013. New Tweets per second record, and how. Twitter Engineering Blog, 16.
[9] Zhang, X., Li, Z., Zhu, S. and Liang, W., 2016. Detecting spam and promoting campaigns in Twitter. ACM Transactions on the Web (TWEB), 10(1), p.4.
[10] Lee, K., Caverlee, J., Cheng, Z. and Sui, D.Z., 2013. Campaign extraction from social media. ACM Transactions on Intelligent Systems and Technology (TIST), 5(1), p.9.
[11] Pagliardini, M., Gupta, P. and Jaggi, M., 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint arXiv:1703.02507.
[12] Mueller, J. and Thyagarajan, A., 2016, March. Siamese recurrent architectures for learning sentence similarity. In Thirtieth AAAI Conference on Artificial Intelligence.
[13] Wu, T., Wen, S., Xiang, Y. and Zhou, W., 2018. Twitter spam detection: Survey of new approaches and comparative study. Computers & Security, 76, pp.265-284.
[14] Aires, R., Manfrin, A., Aluisio, S. and Santos, D., 2004. Which classification algorithm works best with stylistic features of Portuguese in order to classify web texts according to users' needs?

[15] Wang, D., Navathe, S.B., Liu, L., Irani, D., Tamersoy, A. and Pu, C., 2013, October. Click traffic analysis of short url spam on twitter. In 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (pp. 250-259). IEEE.

[16] Tang, D., Wei, F., Qin, B., Liu, T. and Zhou, M., 2014, August. Coooolll: A deep learning system for twitter sentiment classification. In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014) (pp. 208-212).

[17] Song, J., Lee, S. and Kim, J., 2011, September. Spam filtering in twitter using sender-receiver relationship. In International workshop on recent advances in intrusion detection (pp. 301-317). Springer, Berlin, Heidelberg.

[18] Yang, C., Harkreader, R. and Gu, G., 2013. Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Transactions on Information Forensics and Security, 8(8), pp.1280-1293.

[19] Benevenuto, F., Magno, G., Rodrigues, T. and Almeida, V., 2010, July. Detecting spammers on twitter. In Collaboration, electronic messaging, anti-abuse and spam conference (CEAS) (Vol. 6, No. 2010, p. 12).

[20] Stringhini, G., Kruegel, C. and Vigna, G., 2010, December. Detecting spammers on social networks. In Proceedings of the 26th annual computer security applications conference (pp. 1-9). ACM.

[21] Wang, A.H., 2010, July. Don't follow me: Spam detection in twitter. In 2010 international conference on security and cryptography (SECRYPT) (pp. 1-10). IEEE.

[22] Chen, C., Zhang, J., Xiang, Y. and Zhou, W., 2015, April. Asymmetric self-learning for tackling twitter spam drift. In 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 208-213). IEEE.

[23] Jin, X., Lin, C.X., Luo, J. and Han, J., 2011, August. Socialspamguard: A data mining-based spam detection system for social media networks. In Proceedings of the international conference on very large data bases.

[24] Liu, S., Zhang, J. and Xiang, Y., 2016, May. Statistical detection of online drifting twitter spam. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (pp. 1-10). ACM.

[25] Lee, K., Caverlee, J. and Webb, S., 2010, July. Uncovering social spammers: social honeypots+ machine learning. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 435-442). ACM.

[26] Chen, C., Zhang, J., Xie, Y., Xiang, Y., Zhou, W., Hassan, M.M., AlElaiwi, A. and Alrubaian, M., 2015. A performance evaluation of machine learning-based streaming spam tweets detection. IEEE Transactions on Computational social systems, 2(3), pp.65-76.

[27] Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.