IOP Conf. Series: Materials Science and Engineering **719** (2020) 012063 doi:10.1088/1757-899X/719/1/012063

Deepbot: A Deep Neural Network based approach for Detecting Twitter Bots

Linhao Luo[1], Xiaofeng Zhang[1], Xiaofei Yang[1] and Weihuang Yang[1]

Department of Computer Science Harbin Institute of Technology Shenzhen, China
luolinhao@stu.hit.edu.cn, zhangxiaofeng@hit.edu.cn, yangxiaofei@stu.hit.edu.cn, yangweihuang@stu.hit.edu.cn

Abstract. Social networks have played a very critical role in very aspect of our daily life. However, a wide variety of bots have been found which are designed for some malicious purposes such as spreading spam mes- sages and faking news. Although various techniques have been proposed, this task is still challenging if we want to judge whether the tweets are posted by a bot or not merely based on the textual information. For this challenge, the Deepbot is designed which adopts the Bi-LSTM model to analyze tweets and a Web interface is provided for public access which is developed using Web service. From our empirical studies, this system can achieve better classification accuracy.

1. Introduction

Social networks, e.g., Twitter and Facebook, have played a more and more important role in our daily life. Users prefer to share their information via Social networks and are susceptible to the messages posted by other users. This naturally results in the pervasive of social media bots which keep on forwarding messages or faking news. This phenomenon is particularly serious for Twitter. Generally, Twitter users could post millions of *tweets* including textual messages and other rich format messages such as images and videos per day. According to [10], it is estimated that around 48 million registered Twitter users are bots. Some of these bots simply forward news and automatically update its status. On the contrary, a majority of these bots keep on spreading spam messages or fake news which may cause serious consequence such as misleading the political election.

To detect such bots, various techniques have been proposed in the literature. However, this task is still challenging. Actually, bots with fixed patterns such as regularly updating status or forwarding messages could be easily detected. However, some bots try to mimic the behaviour of human beings and thus are hard to detect. This attracts more and more research efforts from both the industry and the academic. Some commercial websites like Botcheck[1 https://botcheck.me/] detects account related features and tries to discover the outliers from these features such as Join date, follower count, tweeting rate, retweeting rate, and tweet text. Bot Sentienl[https://botsentinel.com] identifies bot based on the inappropriate activities of bots. [11] Extracts several graph-based features as well as some content-based features to detect the bots sending spam messages.

Apparently, this task is challenging as the latest bots are designed to mimic human beings to post tweets. Thus, the semantic content of tweets posted should be analyzed. In this paper, we propose a Bi-LSTM based Web application which can quickly detect bots based on only one piece of tweet. Based on the analysis results, the system outputs the probability that whether the input tweet is posted by a bot or not.

2. Related Works

Social network bot detection has long been studied in the literature [1, 2, 4, 9]. Most of existing works focus on the account level classification. In these approaches, the abnormal accounts are detected as they may share similar features like email address and account creation datetime [6]. Some works found that bots usually post tweet using automatic devices with hijacked IP address [1]. Similarly, their social ties as well as their topological network structure could be used to detect abnormal accounts.

On the contrary, some researchers treat the bot detection issue as a classification task. In these works, a number of features are designed for the classification task which are the number of followers or friends, the ratio of the friends to followers, the percentage of bi-directional friends, and the standard deviation of unique numerical IDs of followers and friends [7]. Authors in [11] proposed a social graph using trending topics, replies and mentions to build the classifier. In, the authors utilized the synchronization features of bots for detection. And [5] adopted the deep learning approach to detect bots. In their work, a basic version of LSTM model is built and both the account information and the tweets' contents were used as the model input. Although this work could achieve comparably good results, they need a large amount of labeled tweets to train model which involves a high manual labeling cost. Alternatively, we propose this new technique which is Bi-LSTM based approach and it only requires a small number of training data. This proposed approach can save the labeling cost and can achieve better classification results.

3. The Proposed Deepbot

The proposed Deepbot contains two components: a trained Twitter bot classifier and a Web interface developed using Web service for public access.

3.1. Twitter Bot Classifier

The Twitter bot classifier is proposed based on a deep neural network model to determine whether the input tweet is posted by a bot or not. To represent the textual features of tweets, First, we embed the tweets into vectors using the Global Vectors for Word Representation (GloVE) [8]. This pre-trained word embedding matrix is denoted as $E \in \mathrm{R}^{|e|\times|V|}$, where $|e|$ denote the length of each word after embedding and $|V|$ is the total number of vocabulary V. Let D denote the number of words in the i-th tweet (Si). Then, each tweet Si could be embedded as a matrix $S\ \mathrm{R}^{D\times|e|}$ by replacing all the words with the corresponding word vector v in E.

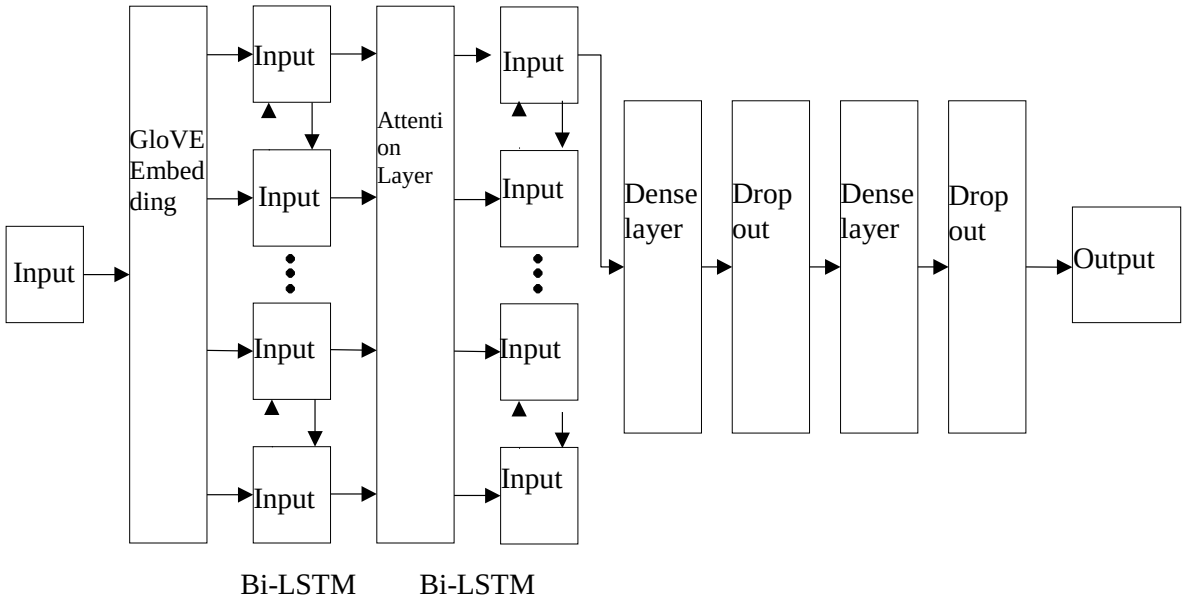

Figure1. The architecture of the proposed Deepbot.

The proposed Deepbot customizes the Bi-directional Long Short Term Memory (Bi-LSTM) [12] to analyze the input tweets and automatically extracts the important textual features. By doing so, it can largely save the manual cost for feature selection and is helpful to build a robust feature space for the learning of a more accurate classifier. For each tweet matrix S, it consists of D word vectors and we have $S = [v1, v2, \ldots, vD]$. As shown in Figure 1, S is treated as a sequence and will be fed into the Bi-LSTM as the input. Then, the Bi-LSTM will store both the long term useful historical textual features but also the short term textual features in the cell unit. The first layer of the adopted Bi-LSTM outputs the hidden state $h_{l1} \in R^{D\times 2u}$, u is the unit number of the single Bi-LSTM cells. And the $hl1$ is then sent to an attention layer which can calculate the importance probability P for each word. This attention mechanism can help the network focus on more important words. Then, the $h_{l1} \times P_i$ is sent to the second Bi-LSTM layer. The second layer only outputs the results of the last cell, i.e., $h_{l2} \in R^{1\times 2u}$. The h_{l2} is then inputted to two fully connected layers with the Relu as the active function. To avoid the over fitting issue, two drop layers are placed between the fully connected layer and the final output layer.

To train the Deepbot, the public dataset is adopted and its link is provided [https://pan.webis.de/clef19/pan19-web/author-profiling.html]. This dataset contains 412,000 annotated tweets posted by 2,060 bots and 2,060 humans, respectively. Its training and testing data are already prepared. The training set includes 144,000 tweets posted by bots and 144,000 tweets by human and the testing set includes 62,000 bots' tweets and 62,000 humans' tweets. In the model training process, we choose "Binary Cross Entropy" as the loss function. The optimizer is "Adam" [3] and the batch size is set to 512. The Deepbot is trained on RTX2080. Finally, we got 79.64 accuracy in test set and the ROC is 87.04.

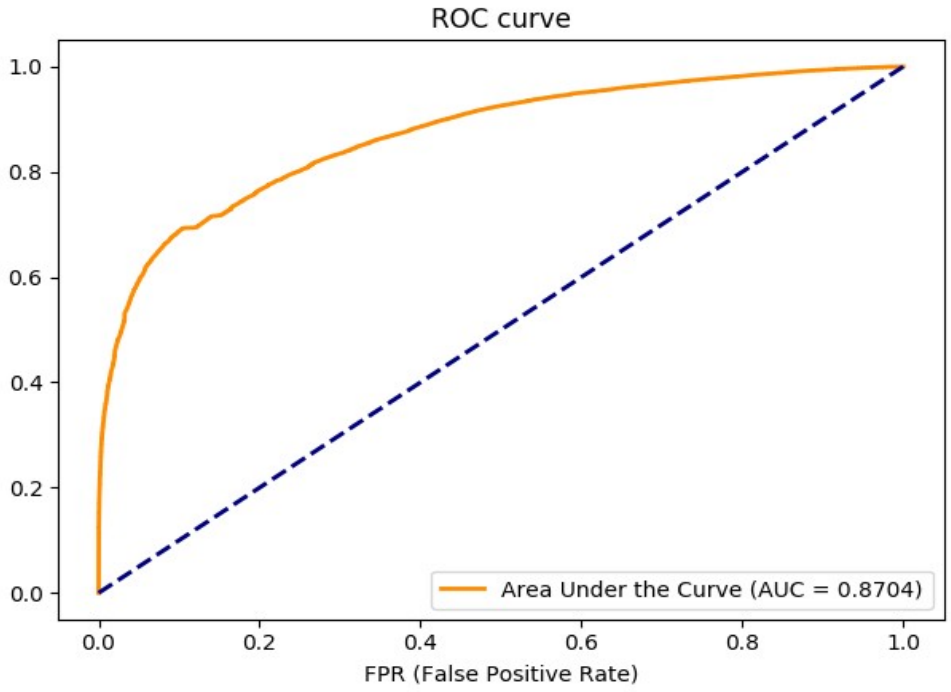

Figure 2. The ROC curve.

3.2. Web Interface

To allow users to access this Twitter bot classifier through the Internet, we provide a public accessible interface developed using Web service. This interface allows a user to upload the tweet and returns the classification result (probability that the tweet is posted by a bot or not) to that user. For the server side, we use Flask, a micro framework developed in Python, to return JSON data generated by the Deepbot.

The working procedure is as follows. After the Web service starts, then the classifier is loaded into the main memory and wait for the message. For the front-end side, we adopt Vue.js to control the data flow, such as receiving user's message, posting the tweets to **the** server, and displaying the returned results.

4. Conclusion

The proposed Deepbot consists of two components which is the bot classifier and the Web interface. In the near future, we will further enhance the model classification ability by designing a more sophisticated deep neural network structure and try to make the Web system of Deepbot to support the high concurrency access.

5. Acknowledgments

This paper is partially supported by Shenzhen Science and Technology Program under Grant No.JCYJ20170811153507788, and the Guangdong Province Science and Technology Department Project under Grant NO.2017B090901022. This work is also partially supported by the National Science Foundation of China under grant No.61872108.

6. References

[1] Cook, D.M., Waugh, B., Abdipanah, M., Hashemi, O., Rahman, S.A.: Twitter deception and influence: Issues of identity, slacktivism, and puppetry. Journal of Information Warfare 13(1), 58–71 (2014)
[2] John, J.P., Moshchuk, A., Gribble, S.D., Krishnamurthy, A., et al.: Studying spam- ming botnets using botlab. In: NSDI. vol. 9 (2009)
[3] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
[4] Krebs, B.: Twitter bots drown out anti-kremlin tweets. Krebs on Security 11 (2011)
[5] Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Information Sciences 467, 312–322 (2018)
[6] Lee, K., Eoff, B.D., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on twitter. In: Fifth International AAAI Conference on Weblogs and Social Media (2011)
[7] Morstatter, F., Wu, L., Nazer, T.H., Carley, K.M., Liu, H.: A new approach to bot detection: striking the balance between precision and recall. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 533–540. IEEE (2016)
[8] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
[9] Stone-Gross, B., Holz, T., Stringhini, G., Vigna, G.: The underground economy of spam: A botmaster's perspective of coordinating large-scale spam campaigns. LEET 11, 4–4 (2011)
[10] Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A.: Online human- bot interactions: Detection, estimation, and characterization. In: Eleventh inter- national AAAI conference on web and social media (2017)
[11] Wang, A.H.: Machine learning for the detection of spam in twitter networks. In: International Conference on E-Business and Telecommunications. pp. 319–333. Springer (2010)
[12] Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase- level sentiment analysis. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005)