Machine Learning Solutions to Online Twitter Bot Detection: Implementation, Comparison and Improvement

Stephen Ling, jling9@wisc.edu

Wyatt Meng, jmeng36@wisc.edu

Bryce Chen, ychen2229@wisc.edu

1. Overview

In this project, we aim to detect whether a Twitter account is human or bot based on their Tweets' content. The dataset we use is a 120,000 rows tabular dataset with columns: Covid-19 related content of the Tweet (English), Twitter ID, and indicator label (human or bot). Feature extraction methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Bidirectional Encoder Representations from Transformers (BERT) would be used. Besides, we plan to compare the model performances of Naive Bayes and RNN. Next, we will optimize model performances by fine-tuning, cross-validation and grid search. Finally, the results will be interpreted in the context of the Covid-19 outbreak and visualized through plots & tables.

2. Background

In general, Nature Language Processing classification tasks involve 4 stages: 1) Preprocessing, 2) Vectorization, 3) Model Fitting, 4) Result Evaluation. In this section, we will discuss commonly used techniques at each stage with focus on research on social bots' detection.

2.1 Preprocessing

There are several commonly used techniques for the preprocessing stage. Stop Words Removal, which is removing commonly used but not informative words like articles and pronouns, can give more focus to the important information in the text (Vijayarani, Ilamathi, & Nithya, 2015). Moreover, Tokenization is used to break the whole text into small chunks to make it easier to assign meaning (Solangi et al., 2018). Also, Stemming directly converts words into root form to reduce inflectional forms of words in the text, and the commonly used models are Affix Removal Stemming, N-gram Stemming, Table Lookup Stemming, etc. In comparison, Lemmatization converts various inflected forms of words into meaningful forms based on the consideration of context (Asghar et al., 2014). Besides, since punctuation marks could not provide any information for analysis, they are removed through techniques named Punctuation Marks Removal (Etaiwi & Naymat, 2017). Finally, some other fancy preprocessing techniques like Part of Speech (POS) tagging are used by researchers to classify words into specific morphological categories (Asghar et al., 2014).

2.2 Vectorization

After the preprocessing stage, it is impossible to directly feed a chunk of text into the classification models. Vectorization would help to extract vectors from the text so that models could use extracted vectors for training and classification. In fact, vectorization could be done by

statistical approaches and deep learning. The traditional vectorization methods commonly used are Term Frequency-Inverse Document Frequency (TF-IDF), Chi-square clustering method, and Latent semantic analysis (LSA) (Liang et al., 2017). Moreover, some neural network-based models like Word2Vec and Doc2Vec are also used for vectorization (Singh & Shash, 2019). For the detection of social bots topics, some researchers use Global Vectors (Glove) and Embedding from the language model (ELMO) for vectorization (Heidari, Jones, & Uzuner, 2020). Also, some researchers use Bidirectional Encoder Representations from Transformers (BERT) which is commonly used for sentiment features extraction (Heidari & Jones, 2020).

2.3 Model Fitting

Based on the literature, the social network of users, profiles of users, account usage, and Twitter content are commonly used by researchers studying the detection of social bots. In most relevant research, researchers tend to use content, users' profiles, and account usage to predict whether the account is a social bot (Rodríguez-Ruiz et al., 2020). However, some researchers only focus on the content and after preprocessing the texts, they use a Recurrent Neural Network (RNN) model with word embeddings to detect bots without using other features like users' profiles or social networks (Wei & Nguyen, 2019). Moreover, some researchers introduce new features like the sentiment of Twitter to analyze whether the content is from humans or bots (Dickerson, Kagan, & Subrahmanian, 2014).

Other than commonly used features for detection of bots' Tweets, the widely used machine learning methods for the detection of social bots on Twitter include both neural network and non-neural network models: Naive Bayes (NBC), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-Term Short-Term Memory (LSTM), etc. (Alothali et al., 2018). Also, if there is no human-bot indicator label for the given text, unsupervised learning methods, especially clustering methods like DenStream, StreamKM++, etc. are used (Khan et al., 2016).

2.4 Performance Evaluation

Finally, since the detection of social bots on Twitter is a classification problem, the Confusion Matrix with Precision, Recall, Accuracy, F-measure calculated, ROC, and AUC are commonly used to evaluate the performance of models. Moreover, some other measurement techniques like Random Walk, Counting Credits at vertex, etc. are also used for the purpose of model performance measurement (Alothali et al., 2018).

3. Statement of Work

3.1 Data

We contacted the first author of "Algorithmic Agents in the Hybrid Media System: Social Bots, Selective Amplification, and Partisan News about COVID-19" and were granted permission to use their dataset for this class project (Duan et al., 2022). The dataset contains English tweets about a specific topic, the Covid-19 pandemic. These tweets were collected from Twitter from March 1, 2020, to May 31, 2020 (under the pandemic outbreak) by matching up with a list of 181 keywords about Covid-19. To reduce the computational complexity and time, we randomly partitioned the dataset and used the first 120,000 rows and eight columns of the original one. In general, the quality of this dataset is high, i.e., there are no single missing values nor NAs in the content and handle columns. The feature columns we plan to use will be:

- *content*: Twitter content sent by users/bots.
- *handle*: represents the Twitter account's ID or nickname.

The second last column among the dataset, *SourceatBot7*, is considered the true label, which is a binary classification outcome indicating the Twitter is sent by human or bot.

[*The dataset is uploaded to GitHub, and can be found in the GitHub Repository*]

The previous research on this data has 3 focuses: statistical comparison between humans and bots, the prevalent Twitter topic for bots, and time series analysis. Duan et al. conclude that human Twitter users tended to post slightly more original content and have more comments than bots, and Twitter bots mainly focused on political or societal topics based on the results of the Structural Topic Model (STM) (2022).

3.2 Methodology

In this project, we try to detect whether a Twitter account is a bot or human based on tweets during the outbreak of Covid-19. Using different machine-learning classification models, we investigated an extensive collection of English tweets relating to COVID-19 (120,000 tweets). We would also like to compare whether sentiments would help detect tweets from bots and figure out whether the neural network model (RNN) would have better performance than the traditional classification model for the given data frame.

Firstly, we will perform preprocessing on the Twitter content. In specific, stop words & punctuation marks removal, tokenization, and stemming would be performed during the preprocessing stage. Then, we are planning to try and compare Term Frequency-Inverse

Document Frequency (TF-IDF) and Bidirectional Encoder Representations from Transformers (BERT) for feature extraction.

The main objective for the model fitting part is to employ and compare different machine learning models learned in this course. Specifically, we would like to compare the performance of the neural network and non-neural network models in the detection of social bots on Twitter. Currently, we plan to use Naive Bayes (NBC) and Recurrent Neural Network (RNN) for classification.

We will train models using 2/3 of the dataset (around 80,000 tweets) and test the models' performances using the rest of 1/3 of the dataset (around 40,000 tweets). Then, we evaluate and compare the model performances through the Confusion Matrix.

We also plan to further optimize the performance of this classification task through finding more optimal combinations of hyperparameters using: Grid Search, and Cross-validation (5 folds).

In the end, we will interpret the results and visualize them through both plots and tables to demonstrate the significance directly.

3.3 Outcome and Performance Evaluation

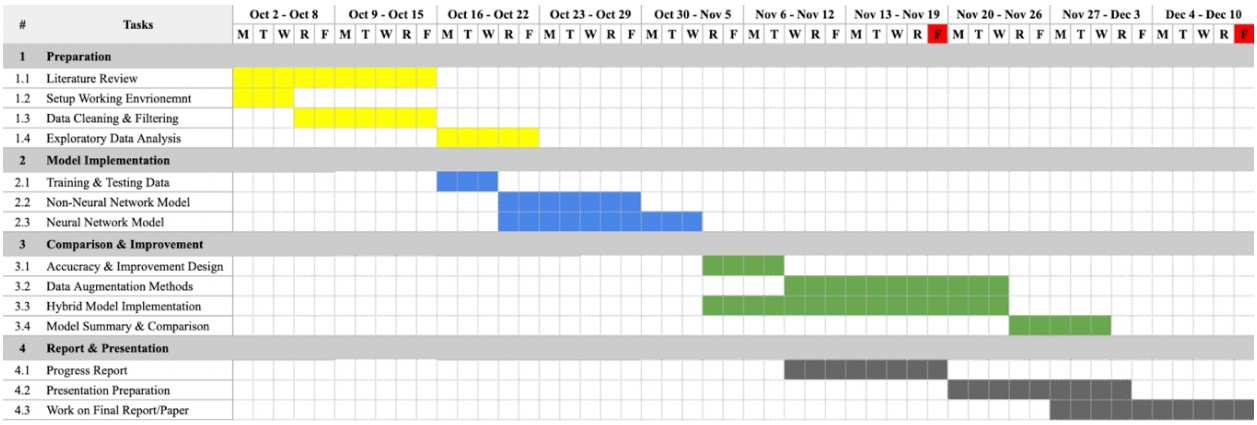
We anticipate correctly applying AI models like machine learning and neural network models in the context of natural language processing. For evaluation, the confusion matrix will be created, and Precision, Recall, F1, and Accuracy are calculated to evaluate and compare the models' performance, conditional on the SourceatBot7 column being the ground truth. Successfully answering the following questions would be considered a successful project:

- Whether extracting sentiments could contribute to the detection of Tweets from Bots?
- Given the same feature vectors, which models would have the better performance (Accuracy) with what parameter settings?

•

4. Project Plan

4.1 Schedule



#	Tasks	Oct 2 - Oct 8					Oct 9 - Oct 15					Oct 16 - Oct 22					Oct 23 - Oct 29					Oct 30 - Nov 5					Nov 6 - Nov 12					Nov 13 - Nov 19					Nov 20 - Nov 26					Nov 27 - Dec 3					Dec 4 - Dec 10				
		M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	**Preparation**																																																
1.1	Literature Review																																															
1.2	Setup Working Envrionemnt																																															
1.3	Data Cleaning & Filtering																																															
1.4	Exploratory Data Analysis																																															
2	**Model Implementation**																																															
2.1	Training & Testing Data																																															
2.2	Non-Neural Network Model																																															
2.3	Neural Network Model																																															
3	**Comparison & Improvement**																																															
3.1	Accucracy & Improvement Design																																															
3.2	Data Augmentation Methods																																															
3.3	Hybrid Model Implementation																																															
3.4	Model Summary & Comparison																																															
4	**Report & Presentation**																																															
4.1	Progress Report																																															
4.2	Presentation Preparation																																															
4.3	Work on Final Report/Paper																																															

4.2 GitHub Links

GitHub Link: https://github.com/JiaheLing/Bots_Human_Detection_with_NN

5. References

Alothali, E., Zaki, N., Mohamed, E. A., & Alashwal, H. (2018, November). Detecting social bots on twitter: a literature review. In *2018 International conference on innovations in information technology (IIT)* (pp. 175-180). IEEE.

Asghar, M. Z., Khan, A., Ahmad, S., & Kundi, F. M. (2014). A review of feature extraction in sentiment analysis. *Journal of Basic and Applied Scientific Research, 4*(3), 181-186.

Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014, August). Using sentiment to detect bots on twitter: Are humans more opinionated than bots?. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)* (pp. 620-627). IEEE.

Duan, Z., Li, J., Lukito, J., Yang, K. C., Chen, F., Shah, D. V., & Yang, S. (2022). Algorithmic Agents in the Hybrid Media System: Social Bots, Selective Amplification, and Partisan News about COVID-19. *Human Communication Research*.

Etaiwi, W., & Naymat, G. (2017). The impact of applying different preprocessing steps on review spam detection. *Procedia computer science, 113*, 273-279.

Heidari, M., & Jones, J. H. (2020, October). Using bert to extract topic-independent sentiment features for social media bot detection. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0542-0547). IEEE.

Heidari, M., Jones, J. H., & Uzuner, O. (2020, November). Deep contextualized word embedding for text-based online user profiling to detect social bots on twitter. In *2020 International Conference on Data Mining Workshops (ICDMW)* (pp. 480-487). IEEE.

Khan, W., Daud, A., Nasir, J. A., & Amjad, T. (2016). *A survey on the state-of-the-art machine learning models in the context of NLP. Kuwait journal of Science, 43*(4).

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP journal on wireless communications and networking, 2017*(1), 1-12.

Rodríguez-Ruiz, J., Mata-Sánchez, J. I., Monroy, R., Loyola-González, O., & López-Cuevas, A. (2020). A one-class classification approach for bot detection on Twitter. *Computers & Security, 91*, 101715.

Singh, A. K., & Shashi, M. (2019). Vectorization of text documents for identifying unifiable news articles. *International Journal of Advanced Computer Science and Applications, 10*(7).

Solangi, Y. A., Solangi, Z. A., Aarain, S., Abro, A., Mallah, G. A., & Shah, A. (2018, November). Review on natural language processing (NLP) and its toolkits for opinion mining and sentiment analysis. In *2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-4). IEEE.

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks, 5*(1), 7-16.

Wei, F., & Nguyen, U. T. (2019, December). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (pp. 101-109). IEEE.