**Creating a Text-Based Classifier to Distinguish**

**Between Human and Autonomous Twitter Users**

by

Benjamin Hermus

Supervised by

Sarah Zerlikovitz, PhD

A Research Project Submitted to the College of Staten Island

Computer Science Department in Fulfillment of the Requirements for

**CSC 450 Honors Workshop**

# Files

## varol-2017.dat

Contains 2,573 annotated Twitter accounts from the 2017 study conducted by Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini titled "Online Human-Bot Interactions: Detection, Estimation, and Characterization." Each row of data represents a unique Twitter account, with the first column being the account’s unique ID, and the second column being a Boolean value set to 1 if the account is a bot.

## varol-2017\_updated.dat

Contains the updated dataset after the 196 deleted/suspended accounts were removed, for a total of 2,377 accounts.

## data/bot/

Contains 715 files of text tweeted by bots. Each file is named after its account’s Twitter ID, and contains the unique tweets (i.e. not retweets) of the user’s 200 latest tweets. The files are comprised of exclusively the characters tweeted by the user, with no added whitespace at all. The following directories follow these same conventions.

## data/human/

Contains 1,476 files of text tweeted by humans.

## bot\_retweets/

Contains 706 files of text retweeted by bots. This file is not included in the “data/” directory, as it was not used in this project’s experimentation.

## human\_retweets/

Contains 1,476 files of text retweeted by humans. This file is not included in the “data/” directory, as it was not used in this project’s experimentation.

## twitter\_update\_dataset.py

Script used to create the “varol-2017\_updated.dat” file from the “varol-2017.dat” file. The script tries to access each of the accounts included in the original dataset. If it succeeds, it includes the account in the new dataset; otherwise, it prints the error sent by Twitter as well as the ID that raised the exception. In this experiment, 196 accounts were removed by this script.

## twitter\_directory.py

Script used to pull all of the tweet text from the accounts in “varol-2017\_updated.dat” and organize them in a manner suitable for testing with scikit-learn. The result is a directory named “data/”, which contains the folders “data/bot/”, “data/bot\_retweets/”, “data/human/”, and “data/human\_retweets/”. Each of these folders contains files named after their respective account’s Twitter ID, which include the accounts tweets/retweets. Note that the folders containing account retweets were manually removed from the “data/” folder for this project. In this experiment, the timelines of 184 accounts were inaccessible by this script, presumably due to the user’s privacy settings.

## twitter\_tests.py

Script used to conduct the experimentation on the collected data. A support vector machine is used as the neural network that creates the classifier, as it is suggested by scikit-learn to use when working with text data. The classifier is then cross-validated 10-fold. In this experiment, the classifier was able to determine with 78±6% accuracy whether or not an account was a bot or human based entirely off of the text of its tweets.

# Recreation

Using the files provided in this directory, this experiment could be recreated (provided one has the credentials to access the Twitter API).

Open the “twitter\_update\_dataset.py” file and provide the appropriate strings for variables ACCESS\_TOKEN, ACCESS\_SECRET, CONSUMER\_KEY, and CONSUMER\_SECRET. Run the script to create a new “varol-2017\_updated.dat” file. Note that this script may take up to an hour to execute completely due to the rate limiting of the Twitter API.

Open the “twitter\_directory.py” file and provide the credentials needed to access the Twitter API. Run the script to create the “data/” directory. Remove the “bot\_retweet/” and “human\_retweet/” folders from the directory, as they are not used with scikit-learn in this experiment.

Lastly, run the “twitter\_tests.py” file to perform the experiment and view the results.

# Issues Encountered

Many of the challenges faced during this project were related to the creation of the tweets dataset contained in the “data/” directory. The first was handling accounts that have been deleted or suspended since the original study was published. This was resolved with the “twitter\_update\_dataset.py” file, which creates a new file containing only the accounts that are currently active.

Next, when attempted to collect the tweet text from each of the users, it became apparent that only the first bunch of characters were being returned. In order to receive the full text, the value “extended” needed to be passed to the “tweet\_mode” parameter when calling the “twitter.statuses.user\_timeline” method. This allowed for the referencing of a new field, “full\_text,” which included the tweet text in its entirety.

It was then realized that including retweets in the scikit-learn tests would greatly skew the results, as any Twitter user can retweet any other user’s tweet, regardless of automation. To deal with this, original tweets and retweets were sorted into separate directories. When the “twitter\_directory.py” script is reading the tweets of an account’s timeline, it first attempts to access a field of the tweet json that is unique to retweets. If this field is accessible (i.e. it exists), the tweet is sorted as a retweet; otherwise, it is sorted as a tweet.

Before running the tests, an issue was encountered specific to macOS. This operating system creates a “.DS\_Store” file in every directory that exists on the machine, and is currently not viewable by any means provided by Apple. Since scikit-learn uses directories to categorize data, having these meaningless files included in the folders would distort the classifier’s training. In order to view and delete these files, the third-party software Funter was used. Furthermore, to prevent the creation of new “.DS\_Store” files, the command “defaults write com.apple.desktopservices DSDontWriteNetworkStores true” was executed in the terminal. Note that this command seems to only affect newly created directories and not preexisting ones.

Finally, when attempting to vectorize the data into bag-of-words format, some characters—most likely emojis or other symbols—could not be decoded. Such characters were not considered in this experiment by passing the value “ignore” to the “decode\_error” parameter of the vectorizer.

## Results

The classifier trained with the tweets from 715 bots and 1,476 humans proved to be fairly accurate when tested with 10-fold cross validation, returning an accuracy of 78±6%. Considering that this experiment used exclusively the text data of these accounts’ tweets, supervised machine learning, and no other Twitter metadata, these results are sufficiently satisfactory.

## What Was Learned

* Improved familiarity with Python syntax and libraries
* Ability to navigate and make us of Twitter APIs
* Usage of scikit-learn’s text data tools
* A great deal of previous research that has been done in effort to distinguish bots and humans on Twitter
* The time and effort that goes into data collection