Just like humans, different bots can be represented by a spectrum of activity. Some bots are more convincing while others are easily recognizable. Given the complexity of the problem and our time constraint, our team decided not to attempt solving the problem described above, and since people have already solved the problem of identifying bots on the lower end of the spectrum (i.e. easily detectable), our team decided to focus on detecting the bots around the middle of the spectrum.[[1]](#footnote-1) These bots are complex enough to make the problem interesting but not so complex that we won’t have enough time to solve the problem.

Since there aren’t any public datasets comprising of real Twitter users (i.e. human beings) and bots, our first task was to collect Twitter profiles. Omer was responsible for collecting Twitter handles for real users, starting from his account. This was convenient because Omer could easily identify humans in his Twitter network. Eldin manually walked checked lists of bots hosted on botwiki.org and Twitter.com/botally/lists/omnibots/members – a Twitter bot repository. With these handles, we wrote a python script to harvest each users’ profile and posts via the Twitter API. The next step is to clean and format the data to prepare it for classification. Namely, we need to extract the followers count, friends count, favorites count, name, created date, screen name, tweets, etc. Some derived attributes we can also use are tweet frequency, lexical diversity, and sentiment.

Our initial plan for classification is to build a TF-IDF from an account’s tweets and use it for sentiment analysis via a SVM. We can then use an Artificial Neural Network with the extracted and derived features as parameters for the final bot-or-not classification. The reason we plan on using an SVM for sentiment analysis is because SVMs are very good at finding optimal boundaries between the possible outputs and we have access to linear as well as non-linear SVMs. Furthermore, we can easily vectorize the text and pass it to the SVM for classification. Classifying text via other classifiers may take more processing and time. Similarly, we are also using Neural Networks since they too are good at modeling complex data. This is convenient since we can just pass feature values as inputs into the network and get a classification.

1. https://www.youtube.com/watch?v=ic4SagX5RFM [↑](#footnote-ref-1)