LEXICAL SYNTAX ANALYSIS FOR HOSTILE BEHAVIOUR IDENTIFICATION

Akshath Jain

ISEF Research Paper

March 7, 2016

Table of Contents

Lexical Syntax Analysis for Hostile Behaviour Identification

Abstract

As the University of Haifa, Israel finds, 90 percent of all terrorist communication happens through social media, most notably Twitter. Current methods of identifying these hostile accounts are either inaccurate or susceptible to human error. This project is designed to analyze social media feeds with supervised machine learning algorithms in order to classify accounts as being terrorist affiliated. Using two key parameters for account classification, diction and affiliation, allows my program to constantly adapt to changing scenarios and yield high success rates. A successful implementation of this project could be used to silence terrorist groups on the internet, and act as an early warning system for imminent threats.

Introduction

Currently, 65 percent of all adults use social media,[1] and because of this, 90 percent of all terrorist communication happens through social media,[2] most notably through Twitter,[3] a type of social media platform used for microblogging with an active user base of over 300 million and growing.[4] Terrorist communication happens extensively through social media because it serves four main purposes: sharing operational and tactical information, a gateway to other online radical content, a media outlet for terrorist propaganda, and remote reconnaissance for targeting

[1] Perrin, Andrew
[2] Wiemann, Gabriel
[3] Ibid
[4] "Number of Monthly Active Twitter Users World Wide from 1st Quarter 2010 to 4th Quarter 2015 (in millions)."

purposes.[5] Twitter, being the most used platform for this activity, would be the most logical service to begin experimentation upon.

Motivated by the recent terror attacks in Paris and San Bernardino, this report presents an empirically based answer to the following question: how can I limit terrorist activity on social media? Therefore, I hypothesize that analyzing social media feeds with machine learning algorithms can identify accounts as hostile.

Algorithm

The methodology used in my algorithm consisted of two key parameters in order to differentiate accounts: diction and affiliation.

Diction analyzes word frequency and word use to categorize unknown accounts. Using account diction gives the algorithm the advantage of constantly adapting to changing scenarios such as different words used. For example, if in month one, words dog, cat, and tiger were prevalent, the algorithm would identify any accounts containing those words as possible matches, but if in month two, words such as river, tree, and fish were prevalent, the algorithm would consider any account containing river, tree, and fish as having a higher probability of a possible match than accounts containing the words dog, cat, and tiger. Note that is an extremely generalized example used for the sake of explanation and that actual implementation is much more sophisticated. For example, the algorithm takes into account words that are constantly frequent among both positive and negative accounts, such as this and it. This is done through a list of "blacklisted words" which is a list of most common words in the English language and contains words excluded from

[5] Blakes, Jason

analysis. During algorithm training, words that are consistently common between positive and negative accounts are added to this list in order to avoid any miscategorizations.

Affiliation analyzes what accounts are linked to an unknown account, such as accounts that follow or are followed by an unknown account and referenced within the Twitter feed. These accounts are then compared to known accounts collected through sample data examination. For example, if unknown account ‘a’ had references to 16 known accounts, and account ‘b’ had references to only 1 known accounts, the algorithm would classify account ‘a’ as more likely match than account ‘b’. Note that as previously, this is an extremely generalized example used for the sake of explanation and that actual implementation is much more sophisticated. In usage, the behavior of the account is considered. For example, if there are matches to known accounts in an unknown account’s Twitter feed (such as retweets), and in the accounts that the unknown account follows, these are valued as higher than those that occur in the followers list. This is because the former set of links requires a more active participation and effort to do as compared to the latter.

The algorithm that my solution uses can be encapsulated into three major categories: data collection, parsing, and analysis.

First is data collection. Using the Twitter4J library allows me to access the Twitter Application Programming Interface (API) through a Java method call rather than through its native JSON language.[6] This helps ensure compatibility with my program, which is written in Java.

[6] *Twitter4J*

Second is data parsing. This step handles the bundle of data received from the Twitter API call. This data is then parsed in order to remove irrelevant information, such as URLs of images and other visual media (these URLs are stored for possible future extensions to this project).

Third is data analysis. This step was the core of my project: the part that determines whether a given account can be deemed “hostile” or “terrorist-affiliated.” In order to examine the data based on historical precedent (i.e. patterns in a sample data set), I utilized a back propagation artificial neural network (ANN) because the adaptability of this particular machine learning algorithm can be attuned to make accurate generalizations on a given data set. Additionally, the multi-layer architecture of the algorithm allows it to make predications based on a non-linear classification pattern. For efficiency and educational reasons, I decided that it would be in my best interest to design and develop a customized neural network as opposed to implementing one via a library. That being said, the artificial neural network implemented into my program is depicted in figure 3.1.

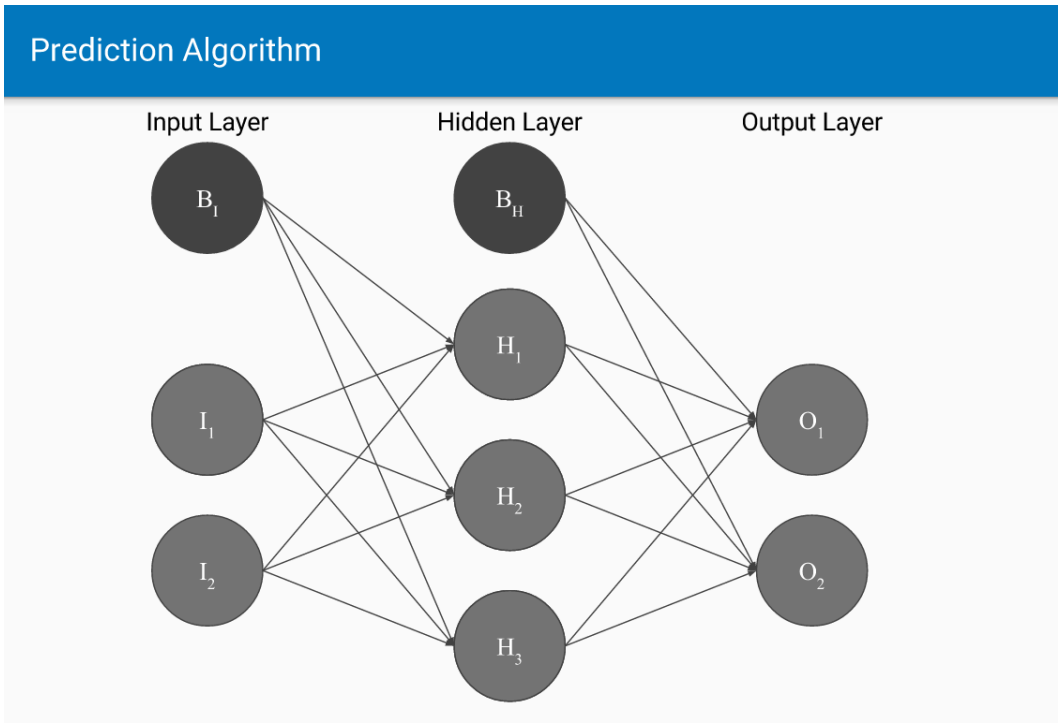

Figure 3.1 – Artificial neural network visualization

Each node on the ANN is representative of a processed/unprocessed data value and each edge is representative of an adjustable weight with a value determined in training (more on this under Experimentation).

From left to right, the first layer is where the data points, diction and affiliation, are inputted, along with a constant bias value of 1, which is designed to prevent output values being indeterminate. The weighted sum of these numbers (including the bias) is then sent to each node on the hidden layer, summarized by:

$$\mathrm{H}_j = \sigma\left(w_{\mathrm{B_I H}_j} + \sum_{i=1}^{2} \mathrm{I}_n w_{\mathrm{I}_i \mathrm{H}_j}\right)$$

$$\sigma(s) = \frac{1}{1 + e^{-s}}$$

Where w_{ij} is the weight from node i to node j.

At the hidden layer, each value is plugged into the sigmoid activation function, as shown in Figure 3.2. It is used as opposed to more rudimentary step functions utilized in perceptrons (a single layer neural network) because it is continuous and therefore differentiable, which is a necessity for training the network.

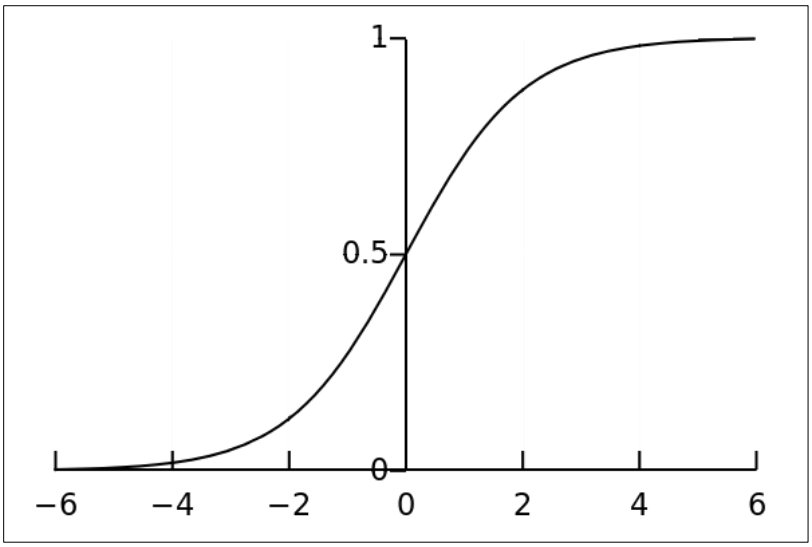

Figure 3.2 – Sigmoid function, $\sigma(s)$

Additionally, the first order derivative can be expressed nicely as:

$$\frac{\partial \sigma}{\partial s} = \sigma(s)\big(1 - \sigma(s)\big)$$

Using the sigmoid function allows each node to mimic a neuron by either firing or not because the $\lim_{s\to\infty} \sigma(s) = 1$ and the $\lim_{s\to-\infty} \sigma(s) = 0$. Although the function looks remarkably like a step function, it is not perfect, especially as it approaches 0, but that is at the cost of using a differentiable function.

These values from the hidden layer are then run through a similar algorithm and outputted to the final output layer, as summarized by:

$$O_k = \sigma\left(w_{B_H O_k} + \sum_{j=1}^{3} H_j w_{H_j O_k} \right)$$

The values in the two output nodes, O_1 and O_2 are then compared, and then the larger value is deemed to be the categorization of the account. Ideally, in a perfect scenario, the difference between O_1 and O_2 should be 1, but in actual implementation, the difference tends to be much smaller.

Experimentation

Experimentation of my solution consisted of two key steps in order to ensure proper functionality: neural network training, and testing.

Before training and testing, I had to gather sample data, and in order to do so, I utilized National Football League (NFL) Twitter accounts as opposed to known terrorist affiliated accounts because I (a) did not want to incite the attention of the NSA, and (b), my solution is designed at classifying accounts as one of two solutions: match or not a match. Because of this, it does not matter whether actual hostile accounts are used for testing because the end result is ultimately the same, either type of account will be affiliated with its own set of accounts and have its own diction style, so if my program can properly identify NFL accounts, in actual implementation, it would be able to identify hostile accounts as well. Additionally, using the official NFL accounts and the fan accounts associated with each of the 32 teams gives me a sufficiently large enough data set to use.

In order to best categorize accounts, the neural network must be trained. This is a process by which the weights are gradually changed to minimize the network error. The back propagation learning routine has five main steps.

1. Every weight in the network is randomly assigned a value between -0.5 and 0.5.
2. Training examples are run, and the error, δO_k, is calculated for the output layer.
3. The error, δH_j, is calculated for the hidden layer using δO_k
4. Weights are adjusted accordingly based on δH_j and δO_k .
5. Repeat steps 2 – 4 until a termination condition is met.

First is initialization. Preceding the first iteration of a neural network, all weights in the system are randomly assigned a value between -0.5 and 0.5. These default values are designed to change through the training process.

Second is output layer error calculation. In this step, a training example *E* is run through the program, and the values at each node are recorded. These values are then processed to determine the output layer error δO_k using:

$$\delta O_k = O_k(E)\big(1 - O_k(E)\big)\big(T_k(E) - O_k(E)\big)$$

Where $T_k(E)$ is the target value for $O_k(E)$ (either 0 or 1).

Third is hidden layer error calculation. The error found for the output layer for example *E*, is then propagated backwards to find the error δH_j, which is calculated using:

$$\delta H_j = H_j(E)\left(1 - H_j(E)\right)\sum_{k=1}^{2} w_{H_j O_k}\delta O_k$$

Fourth is weight adjustment. Using the error calculated at both the hidden and output layers, $\delta \mathrm{H}_j$ and $\delta \mathrm{O}_k$ respectively, the change in weights, $\Delta w_{\mathrm{I}_i\mathrm{H}_j}$ and $\Delta w_{\mathrm{H}_j\mathrm{O}_k}$, are calculated using:

$$\Delta w_{\mathrm{I}_i\mathrm{H}_j} = \eta \mathrm{I}_i(E)\delta \mathrm{H}_j$$

$$\Delta w_{\mathrm{H}_j\mathrm{O}_k} = \eta \mathrm{H}_j(E)\delta \mathrm{O}_k$$

Where η is a learning rate that determines by how much each weight changes; a larger number means more rapid changes, whereas a small number means finer changes. I chose $\eta = 0.1$ as the learning rate as it provides the best balance between rapid change and refinement.

Fifth is repetition. Steps 2 – 4 are repeated until a termination condition is met after each epoch (complete run through the testing data). The training process will end when the accuracy set within the termination condition is met. This accuracy would ideally be 100 percent, but this also would result in the network being overfitted to the sample data set, so to prevent this, a second data set, known as the validation set, is used to test the network. Alarmingly, as shown in Figure 4.1**,** if allowed to train unchecked, the error rate for the training set goes down, but the error rate for the test set starts to go up. Because of that, the termination condition is set to the iteration at which the error rate increases in the validation set, and the weights to the run prior are adopted, but also accommodates for local minima in the validation set.

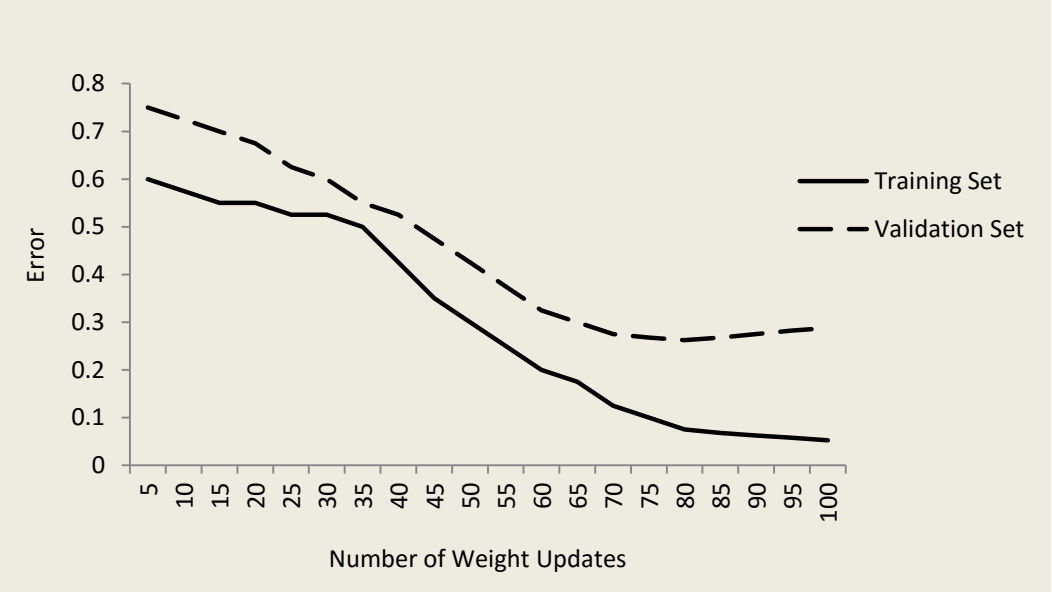

Figure 4.1 – Graph of training set error to validation set error

Testing consisted of three trials with 64 accounts in each trial. The data set was broken up into 32 positive accounts and 32 negative accounts. The values of the accounts were known to the program, but only to determine the accuracy rate of the program at the end of each trial.

Results

A breakdown of the solution implemented proves it to be highly accurate at correctly predicting account types, with 92 percent accuracy. The accuracy was determined by taking the observed over estimated number of accounts correctly identified: $^{176}/_{192} \times 100\%$.

	Trial 1	Trial 2	Trial 3
False Negatives on True Accounts	3	2	3
False Positives on False Accounts	3	3	2

Figure 5.1 – Algorithm Error Rate

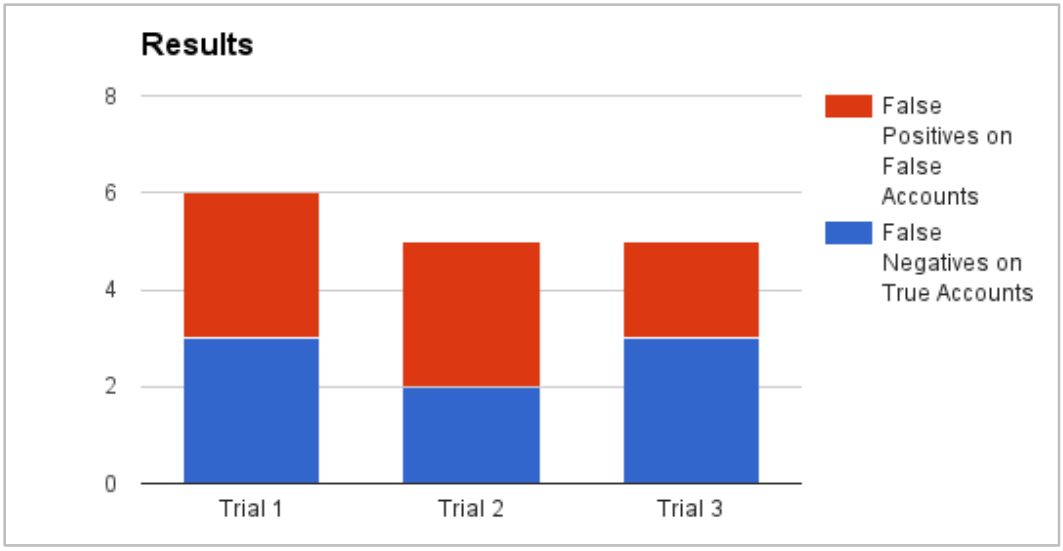

Figure 5.2 – Graph of Algorithm Error Rate

As shown in figures 5.1 and 5.2, the total number of errors across the three trials is on average 5.33, with the false positives and false negatives evenly distributed among the two. There are three possible reasons as to explain these misidentified accounts: the accounts are anomalies, the neural network was optimized to a local minimum, and the neural network was adapted to the nuances of the training set.

The first possible reason is that these misidentified accounts were slight anomalies in the data set, and that these accounts were not in proper threshold for identification, either on account diction or affiliation. In the future, I could address these misidentifications by adding additional parameters and refining upon the ones I currently have in order to minimize the error.

The second possible reason is that the artificial neural network was optimized to a local minimum as opposed to a global minimum. This can be fixed easily by adding a momentum to $\Delta w_{\mathrm{I}_i\mathrm{H}_j}$ and $\Delta w_{\mathrm{H}_j\mathrm{O}_k}$. Like a ball rolling down a hill, this momentum would be a small amount added to $\Delta w_{\mathrm{I}_i\mathrm{H}_j}$ and $\Delta w_{\mathrm{H}_j\mathrm{O}_k}$ based on the last weight adjustment in order to avoid local minima. This would ensure that the network be optimized to a global minimum.

The third possibility is that the neural network was trained too well and was overfitted to the nuances of the training set. This would mean that instead of looking to general trends upon a data set, it is now looking to idiosyncrasies that rely on nuance, and possible outlier, values. This can be easily fixed by using a larger data set during training and expanding upon the neural network architecture for more complex classification patterns.

Discussion

In this report, I attempt to justify an empirically based answer to the following question: how can I limit terrorist activity on social media? In an attempt to answer this question, I created an appropriate hypothesis: analyzing social media feeds with machine learning algorithms can identify accounts as hostile.

In order to provide support for my hypothesis, I created an algorithm that would analyze social media feeds on Twitter to determine if an account was affiliated to a terrorist insurgency. This program, through testing, is shown to be 92 percent accurate in indentifying these hostile accounts, which answers my research question posed and supports my hypothesis because it shows that when implemented, this software could be used to facilitate the appropriate action in order to curb terrorist activity on social media.

The result that I obtained is not unwarranted because previous studies conducted with similar goals in mind provide comparable results. Most notably, Matthew Gerber, a professor at the University of Virginia, created an algorithm to analyze Twitter feeds in an attempt to predict

criminal activity within the United States.[7] In his findings, he concluded that by analyzing spatiotemporal patterns of Twitter activity in relation to Tweet semantics can prove to be more fruitful than current kernel density estimation techniques. This particular study, along with others that use Twitter to predict national election results and disease,[8,9] rely on one key parameter: diction. These studies encapsulate upon the fact that by analyzing key features relevant to the study, a nomothetic statement can be created about any given account, which provides further evidence to support my hypothesis that machine learning algorithms can be used to identify accounts as hostile.

Furthermore, based on the precedent established by the National Institute of Justice, the efficacy of a computer algorithm used for criminal identification purposes is "based on the consequences of errors."[10] Elaborating, a project can be deemed a success if both the societal and financial impacts of errors are minimal. Looking to my project, in the case of an error, an account would slip through the cracks only to be identified by other classification methods already in use.

Although not perfect and with an error rate of 8 percent, this algorithm does support my hypothesis because it demonstrates the ability of machine learning algorithms to correctly classify social media accounts. As discussed previously, the prime candidates for the error rate are anomalies in the account, neural network optimization, or neural network overfitting. Even then, critics may stipulate higher accuracy rates as even a 92 percent accuracy rate would allow certain accounts to slip through the cracks, but it should be noted that this program is currently

[7] Gerber, Matthew
[8] Louis, Connie St, and Gozde Zorlu
[9] Jungherr, Andreas
[10] Ritter, Nancy

intended to be used as a supplement for current terrorist account identification techniques, namely account scrutiny based on user feedback. Even then, the societal implications for this are noteworthy. If implemented properly, this algorithm could act as a method for silencing terrorist groups on the internet, and act as an early warning system for imminent threats.

In future ameliorations to my algorithm, I could enhance upon my current parameters, such as refining the diction analysis algorithms, and add additional parameters, such as visual media analysis. With these future improvements, I could potentially see both accuracy improvements and error reductions.

Conclusion

I can conclusively state that my project provided the empirically evidence needed in determining my hypothesis correct; using machine learning algorithms can identify social media accounts as hostile. I discovered that my algorithm is 92 percent accurate in correctly classifying accounts and that the 8 percent error rate only show room for improvement. In short, this project can be deemed a success, with notable societal benefits.

Works Cited

Blakes, Jason A., ed. *#Terrorist: The Use of Social Media By Extremist Groups*. *Academia*. Academia, Oct. 2014. Web. 26 Jan. 2016. <https://www.academia.edu/9015149/_Terrorist_The_Use_of_Social_Media_By_Extremist_Groups>.

Brooking, E.T. "Anonymous vs the Islamic State." *Foreign Policy*. Foreign Policy, 13 Nov. 2015. Web. 24 Nov. 2015. <https://foreignpolicy.com/2015/11/13/anonymous-hackers-islamic-state-isis-chan-online-war/>.

Chemaly, Soraya. "Twitter's Safety and Free Speech Tightrope." *Time*. Time, 23 Apr. 2015. Web. 24 Nov. 2015. <http://time.com/3831595/twitter-free-speech-safety/>.

Coker, Margaret, Sam Schechner, and Alexis Flynn. "How the Islamic State Teaches Tech Savvy to Evade Detection." *Wall Street Journal*. Dow Jones & Company, 16 Nov. 2015. Web. 24 Nov. 2015. <http://www.wsj.com/articles/islamic-state-teaches-tech-savvy-1447720824>.

Colton, Simon. "Multi-Layer Artificial Neural Networks." *Imperial College London*. Imperial College London, 2004. Web. 6 Mar. 2016. <http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html>.

Gerber, Matthew S. *Predicting Crime Using Twitter and Kernel Density Estimation*. N.p.: n.p., 2014. *Predictive Technology Laboratory @ University of Virginia*. Web. 6 Mar. 2016. <http://ptl.sys.virginia.edu/ptl/sites/default/files/manuscript_gerber.pdf>.

Jungherr, Andreas, et al. *Digital Trace Data in the Study of Public Opinion An Indicator of Attention Toward Politics Rather Than Political Support*. N.p.: Sage Journals, 2014.

Social Science Computer Review. Web. 6 Mar. 2016. <http://ssc.sagepub.com/content/early/2016/02/15/0894439316631043.abstract>.

Louis, Connie St, and Gozde Zorlu. *Can Twitter Predict Disease Outbreaks? The British Medical Journal*. BMJ, 17 May 2012. Web. 6 Mar. 2016. <http://www.bmj.com/content/344/bmj.e2353.abstract>.

Mastroianni, Brian. "Could Policing Social Media Help Prevent Terrorist Attacks?" *CBS News*. CBS News, 15 Dec. 2015. Web. 26 Jan. 2016. <http://www.cbsnews.com/news/could-policing-social-media-prevent-terrorist-attacks/>.

"Number of Monthly Active Twitter Users World Wide from 1st Quarter 2010 to 4th Quarter 2015 (in millions)." *Statista*. Statista, 2015. Web. 6 Mar. 2016. <http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>.

Perlroth, Nicole, and Mike Isaac. "Terrorists Mock Bids to End Use of Social Media." *New York Times*. New York Times, 7 Dec. 2015. Web. 26 Jan. 2016. <http://www.nytimes.com/2015/12/08/technology/terrorists-mock-bids-to-end-use-of-social-media.html>.

Perrin, Andrew. *Social Media Usage: 2005-2015*. *Pew Research Center*. Pew Research Center, 8 Oct. 2015. Web. 6 Mar. 2016. <http://www.pewinternet.org/files/2015/10/PI_2015-10-08_Social-Networking-Usage-2005-2015_FINAL.pdf>.

Ritter, Nancy. "Predicting Recidivism Risk: New Tool in Philadelphia Shows Great Promise." *National Institute of Justice*. Office of Justice Programs, 27 Feb. 2013. Web. 27 Jan. 2016. <http://nij.gov/journals/271/Pages/predicting-recidivism.aspx>.

Secara, Diana. "The Role of Social Media in the Work of Terrorist Groups. The Case of ISIS and Al-Qaeda." *Research and Science Today* 3 (2015): 77-83. Print.

Stergiou, Christos, and Dimitrios Siganos. "Neural Networks." *Imperial College London* 4 (1997): n. pag. Print.

Twitter4J. 4th ed. Vers. 0. Rel. 4. *Twitter4J*. Twitter4J, n.d. Web. 6 Mar. 2016. <http://twitter4j.org/en/index.html>.

Wiemann, Gabriel. *New Terrorism and New Media*. Research rept. no. 2. *Wilson Center*. Wilson Center, 2014. Web. 26 Jan. 2016. <https://www.wilsoncenter.org/sites/default/files/new_terrorism_v3_1.pdf>.