AI Based Opto-Lexical Pattern Analysis for Behavior Categorization

Akshath Jain

11th Grade,
North Allegheny Senior High School,
akshath.r.jain@gmail.com

Abstract. As per the status quo, 90 percent of all terrorist communications occur through social media. This project is designed around ameliorating, or even solving, the aforementioned problem. This is done by analyzing publicly accessible accounts using opto-lexical techniques (images, media, text, etc.) for general behavior categorization, and making distinctions upon those accounts. This allows a custom designed artificial neural network to determine which accounts are potentially hostile in nature. After experimentation, this algorithm was deemed to be 81 percent accurate in classifying accounts. A successful implementation of this project could be used as an early warning system for imminent threats and a potential tool for silencing terrorists on the internet.

Keywords: Social media, neural network, pattern recognition, behavior categorization

Table of Contents

1. Introduction

An alarming increase in terrorist activity on the internet has warranted the need for a solution to eliminate the threat in a timely and cost effective manner. For example, Gabriel Wiemann, a Professor of Communications at the University of Haifa, in Israel, empirically concludes that over 90 percent of all terrorist communication happens through social media, most notable Twitter.[1] This is because terrorist groups, like the Islamic State (also referred to as IS and ISIS) use social media in four primary ways. First is to share operation and tactical information, second is as a gateway to other radical online content, third is as a media outlet for terrorist propaganda, and fourth is for remote reconnaissance for targeting purposes.[2]

As alluded to previously, the majority of this activity occurs on Twitter, which is a microblogging service with an active user base of over 300 million people.[3] This elicits a widespread audience for any category of activity, including terrorism, which is why a recent study by the Brookings Institute concludes that there are an estimated 90,000 terrorist accounts on Twitter.[4] This comprises approximately 0.03 percent of the active user base.

In fact, as shown in Figure 1.1, in the time period from 2012 to 2014, as the number of active users on Twitter grew, so did the number of fatalities due to terrorist attacks.[5] Albeit, I am not implying a correlation between these two phenomena, but it is an interesting trend to note.

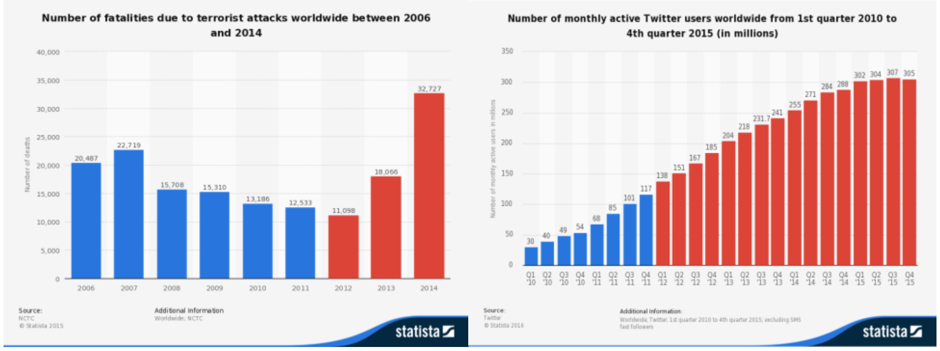

Figure 1.1 - the correlation between the number of active Twitter users and the amount of terrorist attacks.

The need for a versatile algorithm arises especially in lieu of current methods used to identify and remove terrorist accounts: namely using account scrutiny based upon user feedback (i.e. a select group of individuals identify and review accounts that various users have flagged for inappropriate or terroristic behavior). However, this solution is highly susceptible to bias and a substantial amount of

[1] Wiemann
[2] Blakes
[3] "Number of Monthly"
[4] Brooking
[5] "Number of Fatalities"

variance between graders. A systemized algorithm, however, would be able to eliminate, or at least mitigate, this bias, resulting in more accurate identification.

As such, the hypothesis for this project is that analyzing social media feeds with machine learning algorithms, including image and text analysis, can successfully classify behavioral patterns.

2. Algorithm

The algorithm used to create a program to test my hypothesis involved a simple four step process to ensure optimal efficiency: collecting all necessary Twitter account data, parsing said data into usable formats, analyzing the data to determine apparent trends, and finally, using an artificial neural network to predict the probability of the data being associated with a terrorist account (also referred to a hostile account or true account).

In order to most efficiently complete my program, I chose to write in Java. I chose Java because it's well-documented and its object-oriented nature allows for creating complex hierarchical structures, as required for my program, in an efficient manner. Moreover, the compatibility of the language was exceptionally useful as I used multiple libraries in the development of my program.

Step one was collecting all necessary data required for analysis. This included harvesting data from the Twitter servers directly, reading values from JSON (JavaScript object notation), and CSV (comma separated values) file formats. I was able to connect to the Twitter servers by utilizing the Twitter4J library, which allowed my program, written in Java, to connect to the Twitter API, which uses the REST API, to return JSON values.[6] The library was able to connect to the API using OAuth authentication protocols and parse the returned JSON values into Twitter account objects. Reading and parsing data from JSON and CSV file formats directly was required to use data from the various data sets used during experimentation.

The second step was data parsing: formatting all collected data into usable formats for my program. As per the concept of encapsulation, the need to delve into such specifics is not warranted.

The meat of the algorithm lies within the third and fourth steps: data analysis and prediction.

2.1 Data Analysis

Data analysis consisted of four primary categories: diction, affiliation to known accounts, visual media analysis, and miscellaneous features.

[6] *Twitter4J*

Diction constituted of analyzing all text associated with an unknown account (i.e. any account in question of being hostile). The data extracted from this included determining the most common words, the percentage match of these words, the average match distribution, and the most common hashtags used. This was done by determining the frequency of each word and then using an adaptation of Tim Peter's list sort algorithm that uses a stable, adaptive, and iterative implementation of merge sort that usually requires less than nlog(n) iterations given the input set is partially sorted.[7] After sorting, the most common words for the unknown account, along with each word's frequency, are determined.

These features from the unknown account are then compared to the features in a database of all terrorist accounts. From this comparison arise parameters such as the number of common word matches, percentage of common word matches, average match distribution, and most common hashtag matches. The holistic database of known terrorist account features was compiled by taking the data from individual known accounts, and determining the overall number of common word matches, percentage of common word matches, etc. Essentially, each unknown account is compared to the holistic database and the amount of similarity is between the two is mapped and stored for later use.

Additionally, the algorithm accounts for common words that exist in languages, such as [the, be, to] in English and [de, que, no] in Spanish (the algorithm operates independent of language). This is done by comparing the most common words that appear in hostile accounts and normal accounts (non-terrorist affiliated accounts) and eliminating any statistical outliers (generally the top 10 percent). For example, words such as [the, be, to] would be excluded from all calculations because of their overall prevalence in the English language. Furthermore, this process eliminates any other common words that exist between true and false accounts; during algorithm training, words that are consistently common between true and false accounts are excluded from calculations in order to avoid any miscategorizations.

Moreover, using diction gives my algorithm the advantage of being able to adapt efficiently to slight behavioral changes over time. For example, if words such as [cat, dog, fish] were popular during month 1, but words such as [river, tree, plant] were popular during month 2, the algorithm would automatically retrain itself to better identify the latter set as being more probabilistically significant as an account match.

The next feature set gathered is affiliation to known accounts. This category analyzes any communication between an unknown account to a known account by analyzing how many of an unknown account's friends, followers, retweets, and content mentions are linked to known accounts.

[7] "Arrays"

The third feature set looks toward all visual media, such as images. These images are processed in two ways. First is by using the Microsoft Computer Vision API (MCV API) to determine an image's adult score, racy score, autogenerated image caption, number of males, number of females, number of faces, average age, width, height, predominant foreground color, predominant background color, black and white status, clipart status (if the image is a clipart style picture), and vector-style status (if the image is a vector image).[8] My program connects to the MCV API using the REST API and a parseable JSON string is returned once all computations are complete. These values are then compared to the holistic database of known accounts. After using the MCV API, the autogenerated image caption is cross examined using the diction analysis algorithms, albeit this data is stored separately from the text based feature sets.

In addition to the values returned from the MCV API, I utilized Google's TensorFlow library to more accurately determine what was in the most central focal point of the image (the area for the focal point was determined by the MCV API).[9] The TensorFlow library simply provided me with an untrained convolutional neural network to build upon for the purpose of my project. In fact, I had to train this network using the CIFAR-10 dataset of 60,000 32x32 color images.[10] The values determined by the convolutional neural network were weighted higher than those determined by the MCV API as these values focused specifically on the focal point of each image.

The fourth and final feature set looked towards all miscellaneous data, including dates, time and time zones, average number of tweets per day, location, and language. These values were then compared to the holistic database and the similarities and differences were mapped.

2.2 Data Prediction

Prediction was a major aspect of my project: it is the part that determines whether a given account can be deemed hostile or terrorist-affiliated. In order to classify behavioral patterns, 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 predictions 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. The custom artificial neural network implemented into my program is depicted in figure 2.2.1.

[8] "Computer Vision API"
[9] "TensorFlow"
[10] "CIFAR-10"

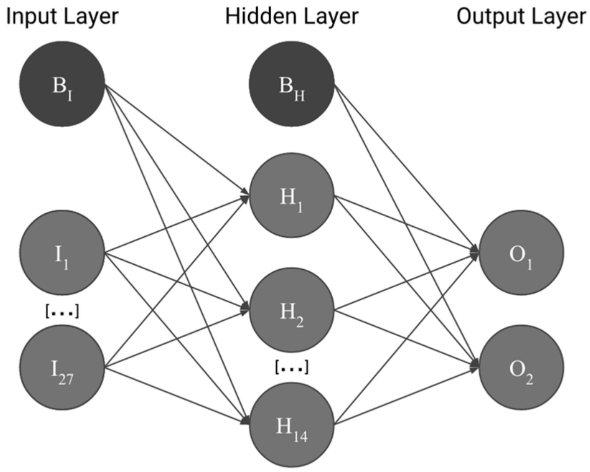

Figure 2.2.1 – Artificial Neural Network diagram

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 feature sets – diction, affiliation, visual media, and miscellaneous – are inputted, along with a constant bias value of 1. The bias values are 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:

$$\mathrm{H}_j = \sigma\left(w_{\mathrm{B_I H}_j} + \sum_{i=1}^{27} \mathrm{I}_n w_{\mathrm{I}_i \mathrm{H}_j}\right)$$

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

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 2.2.2. It is used as opposed to more rudimentary step functions utilized in perceptrons (single layer neural networks) because it is continuous and therefore differentiable, which is a necessity for training the network. The number of nodes on the hidden layer was determined by averaging the number of input nodes and output nodes – this method is generally accepted as yielding the optimal number of nodes for the most efficient training and execution times.

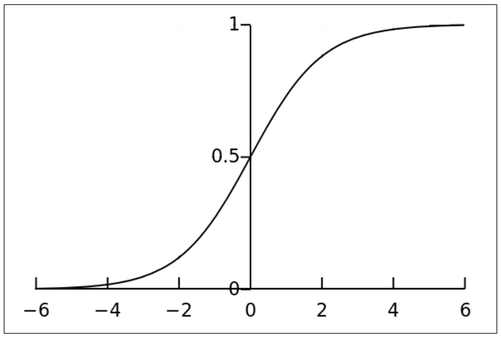

Figure 2.2.2 – Sigmoid function, $\sigma(s)$

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 firing 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 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:

$$\mathrm{O}_k = \sigma\left(w_{\mathrm{B_H}\mathrm{O}_k} + \sum_{j=1}^{14} \mathrm{H}_j w_{\mathrm{H}_j\mathrm{O}_k} \right)$$

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. However, in actual implementation the difference tends to be much smaller.

3. Experimentation

Experimentation consisted of two parts: neural network training and neural network testing. Since the neural network is a supervised machine learning algorithm, I had to provide it with known classifications of accounts. Two data sets were used for training and testing: one with known terrorist accounts and the other with regular Twitter accounts.

The data set with hostile accounts was a publicly available data set of decommissioned ISIS accounts compiled by hand; however, it is important to note that *all* accounts in this data set were no longer in use and had been removed from

Twitter.[11] Regardless, this set was able to provide me with approximately 20,000 Tweets from over 200 known accounts in a JSON format, giving me an ample number of data points to work with.

The second data set used was a publicly available compilation of approximately 40,000 Twitter accounts, broken up proportionally into different categories based on activity.[12] For example, the vast majority of accounts belonged to individuals; however, some belonged to brands and products, companies and organizations, local businesses, movies and television, music, sports, and websites. Even though the data set was proportionally distributed, the accounts were unmarked, so I had to go through, by hand, and mark each account as falling under one of the eight aforementioned categories. This process undoubtedly involved some human-error, but it was the best publicly available data set for the purpose of this project.

3.1 Neural Network Training

In order to 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, $\delta\mathrm{O}_k$, is calculated for the output layer.
3. The error, $\delta\mathrm{H}_j$, is calculated for the hidden layer using $\delta\mathrm{O}_k$
4. Weights are adjusted accordingly based on $\delta\mathrm{H}_j$ and $\delta\mathrm{O}_k$.
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 $\delta\mathrm{O}_k$ using:

$$\delta\mathrm{O}_k = \mathrm{O}_k(E)\big(1 - \mathrm{O}_k(E)\big)\big(\mathrm{T}_k(E) - \mathrm{O}_k(E)\big)$$

Where $\mathrm{T}_k(E)$ is the target value for $\mathrm{O}_k(E)$ (either 0 or 1).

Third is hidden layer error calculation. The error found for the output layer for example E is propagated backwards to find the error $\delta\mathrm{H}_j$, which is calculated using:

[11] Zaman

[12] Wassner

$$\delta\mathrm{H}_j = \mathrm{H}_j(E)\left(1 - \mathrm{H}_j(E)\right)\sum_{k=1}^{2} w_{\mathrm{H}_j\mathrm{O}_k}\delta\mathrm{O}_k$$

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:

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

Where η is a learning rate that determines how much each weight changes; a larger number means more rapid changes, whereas a small number means finer changes. I chose η = 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 ended when the accuracy dictated by the termination condition was 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, was used to test the network. Alarmingly, as shown in Figure 3.2.1**,** if allowed to train unchecked, the error rate for the training set goes down, but the error rate for the validation 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 but decreases in the training set; the weights to the run prior are then adopted. However, the algorithm does accommodate for local minima in the validation set.

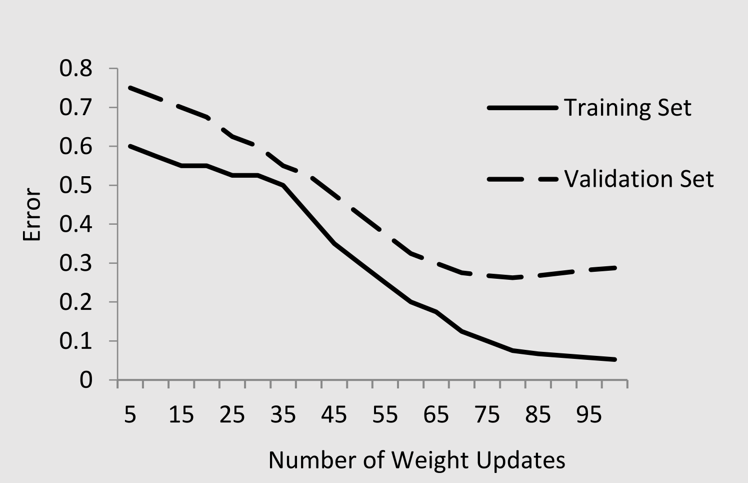

Figure 3.1.1 – Graph of training set error to validation set error

A ratio 0.03 percent of hostile accounts to regular accounts would yield the most accurate real-world-scenario for training. However, due to limited resources (number of sample accounts, time, etc.) I had to scale the proportion of hostile accounts to 1 percent of the entire dataset to best represent a real-world-scenario. This means that for both the training and validation sets, I used 12 positive (hostile) accounts and 108 negative accounts, with the negative accounts broken up into the eight aforementioned categories.

3.2 Neural Network Testing

As with neural network training, the proportion of hostile accounts was once again scaled to 1 percent – a result of possessing limited resources. However, this was still able to represent a real-world-scenario. Testing consisted of nine trials: 64 true hostile accounts and 6,336 negative accounts, with the negative accounts being proportionally split into the eight aforementioned categories. The results were then recorded and the percentage of misclassifications calculated.

4. Results

An analysis of the empirical data shows my algorithm to by 81 percent accurate in correctly identifying accounts. As shown in figures 4.1 and 4.2, the largest percentage of miscategorizations occurred in trial one and trial nine, with 11.0 percent and 8.7 percent error respectively.

	True Accounts	False Accounts							
	Trial 1: Hostile	Trial 2: Brands and Products	Trial 3: Companies and Organizations	Trial 4: Local Businesses	Trial 5: Movies and Television	Trial 6: Music	Trial 7: Sports	Trial 8: Websites	Trial 9: Individual
Percent Misclassifications	11.0%	2.7%	1.0%	0.8%	1.3%	2.2%	1.1%	0.6%	8.7%

Figure 4.1 – Table of percent misclassifications

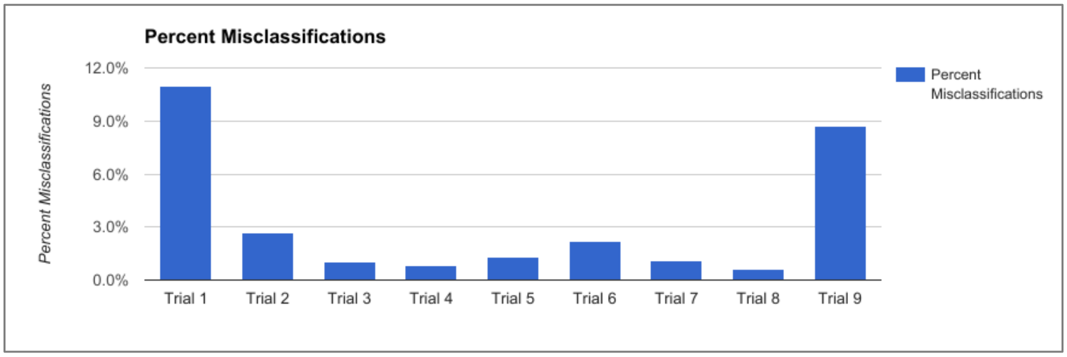

Figure 4.2 – Graph of percent misclassifications

This error most likely resulted from the greatly varying uniqueness that exists in both the terrorist accounts and individual accounts in trials one and nine respectively. For example, two individual accounts (or hostile accounts), controlled by people with completely different habits, hobbies, tastes, and styles, are more likely to be vastly different from one another than two accounts from companies and organizations. This is because most companies and organizations on Twitter – and social media in general – are there to promote their own products, minimizing the amount of differences between the accounts.

However, another probable explanation, albeit less than the previous one, could be that the neural network was either optimized to the local minimum or adapted to the nuances of the training set. If the former explanation is correct (improper optimization), this could be easily fixed by easing the termination conditions during neural network training. However, if the latter explanation proves true (overfitting), then the termination conditions must be strengthened and training set size increased to prevent the neural network from becoming accustomed to the nuances and idiosyncrasies of the training set.

To further examine the efficacy of the algorithm, distributions from three key parameters – text based common word matches, visual media based caption matches, and percent affiliation – were mapped. This is seen in figures 4.3, 4.4, and 4.5 respectively.

A commonality among all three appears when the particulars of the distribution pattern are analyzed: true (hostile) accounts tend to be distributed further left with a greater number of matches or percent affiliation whereas negative accounts tend to be nearer zero. However, when an account lies on areas of overlap, the algorithm I developed tends to misclassify the accounts, as no clear distinctions can be made.

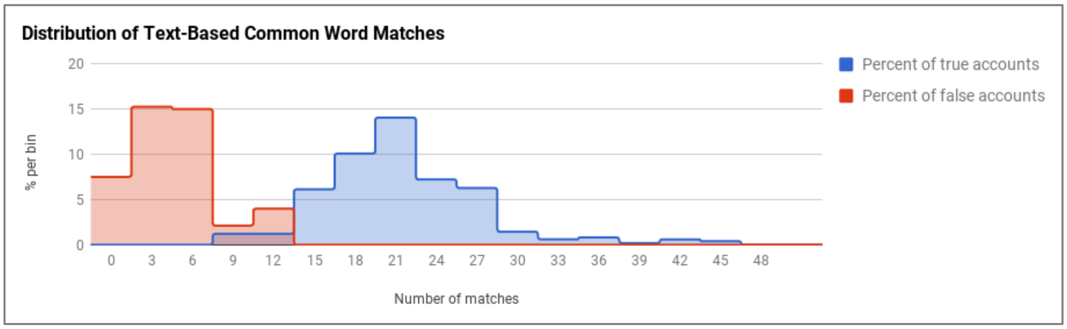

Figure 4.3 – Distribution of Text-Based Common Word Matches

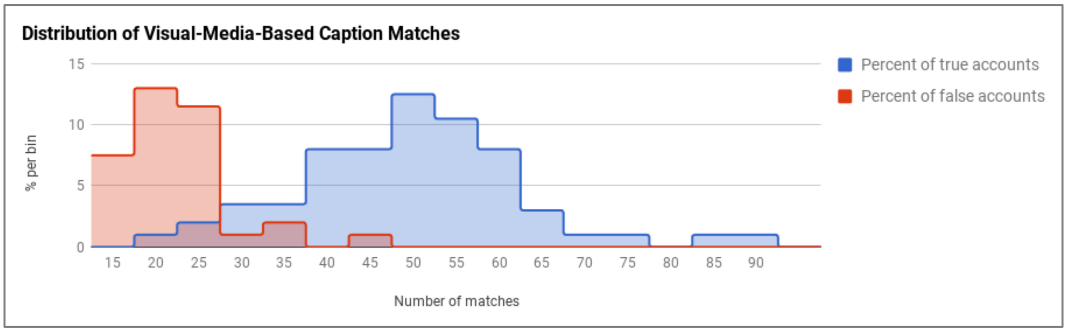

Figure 4.4 – Distribution of Visual-Media-Based Caption Matches

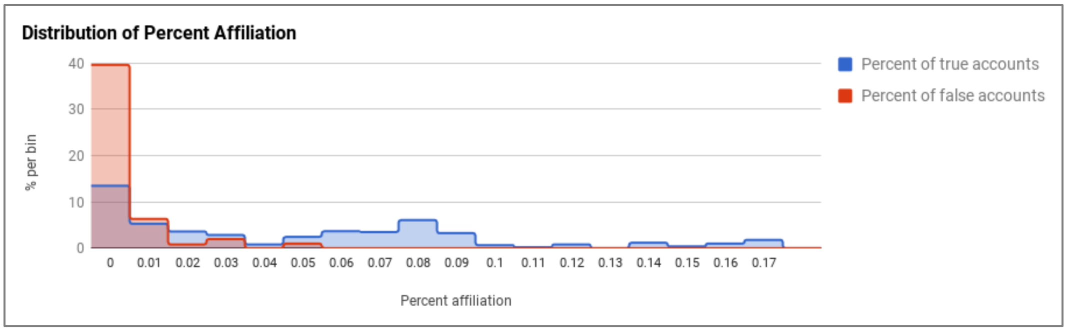

Figure 4.5 – Distribution of Percent Affiliation

5. Discussion

In this report, I attempted to empirically answer the question: how can terrorist activity on social media be limited? As such, I created an algorithm that analyzes a Twitter account's speech, images, interactions, and other miscellaneous data to classify it as a potential hostile account.

As demonstrated through rigorous experimentation, my algorithm was 81 percent accurate in correctly classifying accounts of various types, from decommissioned ISIS accounts to companies, websites, and individuals. And as previously discussed, the error most likely resulted from significant account variance, improper optimization, or overfitting. These could easily be fixed by adding more parameters, loosening the termination condition and adding more accounts to the training set, or strengthening the termination condition respectively. However, by the nature of human behavior, it is extremely difficult to control account variance.

In fact, these observations are not unwarranted. For example, 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.[13] In his findings, he concluded that 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, demonstrates the validity of the results discussed in this paper.[14][15] Furthermore, Facebook, a multi-billion-dollar social media company, recently implemented Artificial Intelligence algorithms, much like my own, to determine if any of their users are suicidal.[16] In fact, Facebook has even started to test these algorithms in the United States.

If this project were to be repeated, I would add additional parameters to minimize the number of misclassifications, test using a larger data set to better represent a

[13] Gerber
[14] Louis
[15] Jungherr
[16] Kelion

real-world-scenario – I had to scale up from 0.03 percent to 1 percent of all active accounts being hostile, and improve upon the efficiency of my algorithm.

6. Conclusion

I can conclusively determine that my experiment was successful and my hypothesis correct: analyzing social media feeds with machine learning algorithms, including image and text analysis, can successfully classify behavioral patterns.

This conclusion is based upon the precedent established by the National Institute of Justice: the efficacy of computer algorithm used for criminal identification purposes is “based on the consequence of errors.”[17] Essentially, if the societal and financial impacts of errors are minimal, the project can be deemed a success. When applied to this project, an error would result in an account slipping through the cracks or being misclassified, only to be identified later by other classification methods in use. As such, the 19 percent miscategorizations rate demonstrates room for improvement in the future.

Even still, some may stipulate accuracy rates higher than 81 percent, as even this would allow certain accounts to slip through the cracks. However, it should be noted that this program is currently intended to be used as a supplement for current terrorist account identification techniques, namely account scrutiny based on user feedback. Regardless, 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.

[17] Ritter

7. Bibliography

"Arrays (Java Platform SE 7)." *Arrays (Java Platform SE 7)*. N.p., 11 Jan. 2016. Web. 01 Mar. 2017. <https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html>.

Blakes, Jason A., ed. *#Terrorist: The Use of Social Media By Extremist Groups*. *Academia*. Academia, Oct. 2014. Web. 13 Feb. 2017. <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. 13 Feb. 2017. <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. 26 Jan. 2017. <http://time.com/3831595/twitter-free-speech-safety/>.

"CIFAR-10 and CIFAR-100 Datasets." CIFAR-10 and CIFAR-100 Datasets. University of Toronto, n.d. Web. 01 Mar. 2017. <https://www.cs.toronto.edu/~kriz/cifar.html>.

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. 13 Feb. 2017. <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. 13 Feb. 2017. <http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture13.html>.

"Computer Vision API." *Microsoft Cognitive Services*. Microsoft, n.d. Web. 01 Mar. 2017. <https://www.microsoft.com/cognitive-services/en-us/computer-vision-api>.

Gerber, Matthew S. *Predicting Crime Using Twitter and Kernel Density Estimation*. N.p.: n.p., 2014. *Predictive Technology Laboratory @ University of Virginia*. Web. 13 Feb. 2017. <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. 13 Feb. 2017.

<http://ssc.sagepub.com/content/early/2016/02/15/0894439316631043.abstract>.

Kelion, Leo. "Facebook Artificial Intelligence Spots Suicidal Users." *BBC News*. BBC, 01 Mar. 2017. Web. 01 Mar. 2017. <http://www.bbc.com/news/technology-39126027>.

Louis, Connie St, and Gozde Zorlu. *Can Twitter Predict Disease Outbreaks? The British Medical Journal*. BMJ, 17 May 2012. Web. 13 Feb. 2017. <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. 13 Feb. 2017. <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. 26 Jan. 2017. <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. 13 Feb. 2017. <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. 26 Jan. 2017. <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. 13 Feb. 2017. <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.

"TensorFlow." *TensorFlow*. TensorFlow, n.d. Web. 01 Mar. 2017. <https://www.tensorflow.org/>.

Twitter4J. 4th ed. Vers. 0. Rel. 4. *Twitter4J*. Twitter4J, n.d. Web. 26 Jan. 2017. <http://twitter4j.org/en/index.html>.

Wassner, Hubert. "Twitter Friends." Twitter Friends. Kaggle, Aug. 2016. Web. 26 Jan. 2017.

Wiemann, Gabriel. *New Terrorism and New Media*. Research rept. no. 2. *Wilson Center*. Wilson Center, 2014. Web. 13 Feb. 2017. <https://www.wilsoncenter.org/sites/default/files/new_terrorism_v3_1.pdf>.

Zaman, Khuram. "How ISIS Uses Twitter." How ISIS Uses Twitter. Kaggle, May 2016. Web. 26 Jan. 2017.