**Sentiment Analysis of Twitter data on Gun Laws in the United States**

**BY: KIRAN KUMAR MUMUDALA**

**Summary:**

Gun related violence is known to be a complex issue and accounts for a large proportion of violent incidents. In this project, we set out to investigate the sentiments expressed by public on a social media platform, namely Twitter, in response to the Texas Church Massacre that occurred on November 5th 2017.

Social media has become a huge platform where people freely express their opinions. Mass shootings often catalyze an active debate and legislation on gun laws in U.S. In this project, we would present the analysis of Twitter data which provides useful insights on this topic. The support and opposition of gun laws is measured or estimated by analyzing over three hundred thousand tweets that have been successfully retrieved from twitter.

Keywords: gun laws, texas massacre, texas shooting, gun control

**Introduction:**

The project mainly focuses on the topic “Gun laws in the United States”. The Gun laws in U.S are a major public issue that has polarized the U.S society. The Gun laws and its effect on mortality have always been under scrutiny.The right to keep and bear arms in the United States is protected by the Second Amendment to the U.S Constitution. But this law is facing a lot of criticism.

The gun laws and its effect on Social media have a wide range of data available on itself about the public opinion on gun laws. With the recent Texas shooting, public opinion played a major role throughout this time period, where discussions of gun control on social media rose in prominence and prevalence. We present an analysis of gun related Twitter data from the recent Texas church massacre, over three hundred thousand tweets that we have retrieved in total.

**Data Collecting:**

Now that we had our idea and knew what we want to look for but we needed to find a way of pulling that data from Twitter. After doing research on the subject we found out Twitter offers a public API to pull data from Twitter feeds. To use the REST API Twitter offered you start by creating a Twitter account then creating your application at apps.twitter.com. Twitter’s documentation was very handy and easy to follow. Once you have done these steps you are ready to call your new application using any programing language listed on Twitter’s developer’s page website. We went with Python being a language simple to use and free to use. The package we used is called tweepy and it was also listed on Twitter developer’s page. A problem arose when we wanted to pull all relevant records from the API but found out you can only call the API so many times in a fifteen minute window. After some more searching we came up with some code to help on with the problem. It was from the blog [www.karambelkar.info](http://www.karambelkar.info). It was used as our code base to pull data from the API and not exceed its call limits. Exceeding call limits too many times would lead to a ban of your Twitter application. We changed the code to write to CSV format over the JSON file format it was built with. It was already great code and did everything we needed it to do so there was not much else to change. Once we had done some testing we ran the Python Script for all our keywords and saved each set of data as its own CSV file ready for testing and use. Once the files were created we had the data we needed. The REST API search allowed us to go back about 2 weeks’ worth of data. View appendix 1 for a sample of our code. Also, the API downloaded around 100 records per second. Overall, it was a great way to pull data from Twitter for our project.

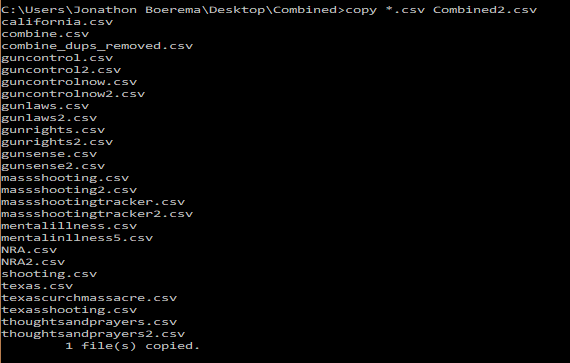
**Data Cleansing:**

For this project we did data cleansing in a few steps. This was because we needed to process the data quickly and not spend tons of time coding. If we continue to collect more data, it will be better to write a python script to do the following steps all in one program and not three different programs to clean the data.

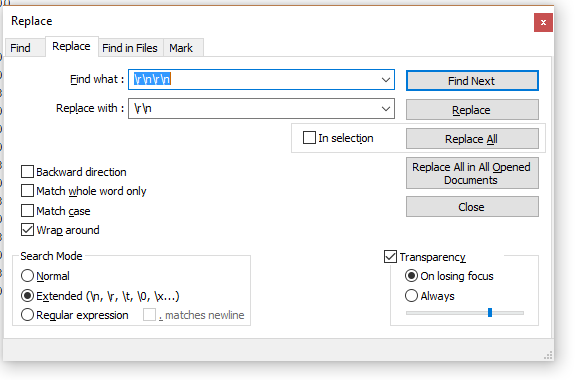
The first thing we had to do was to join all the separate data files we collected from different Twitter hashtags together. We wanted three different sets of data. One dataset containing the first set of data was collected on the Texas Church shooting. The second data collected was on the California School shooting in November 2017. The last set of data was a combined set of both sets of data.

Now that we picked what our datasets were going to be, we then had to clean them. The first step was to group combine all the data files into our three data sets. We did this by using Windows Command Prompt. We used windows “copy” function to create a new file looking for all CSV files in our file folders. Example: copy \*.csv Combined.csv. This command then copied all the files that were named wildcard ‘csv’ and built a new file called Combined.

Example:

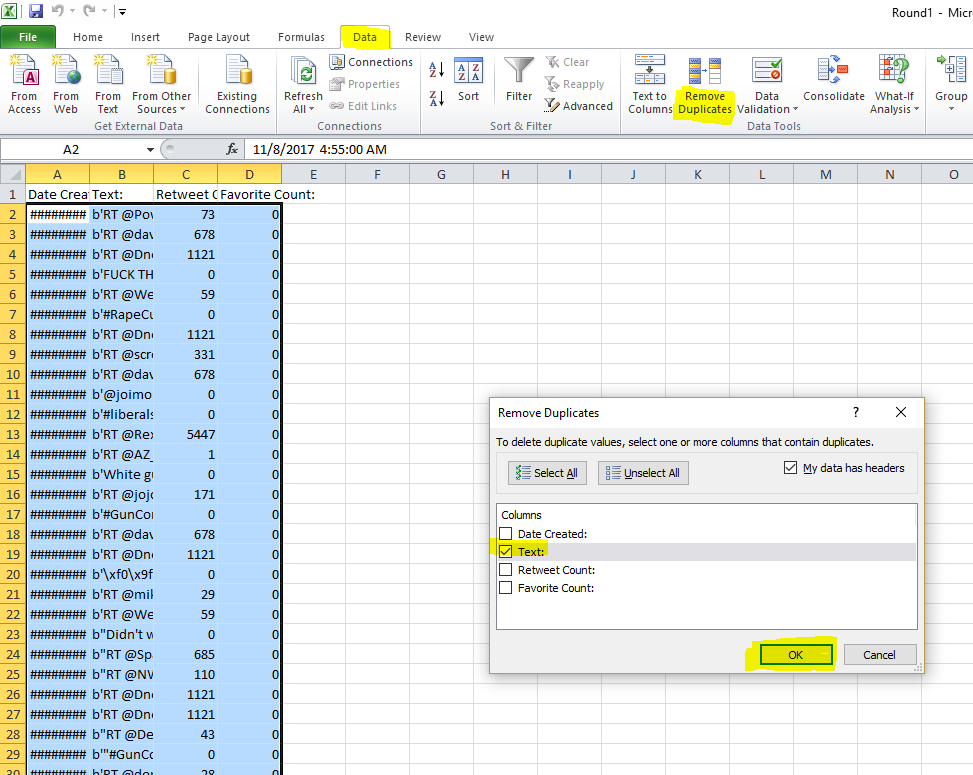


After getting the three csv files that we needed we had to remove some data. The data that was removed was blank rows that were created when each collection csv file was created and then duplicates of Tweets in each set of data. To remove the blank rows we used NotePad++ because the combined data with blank rows where too big for 32bit excel. We simply did a “find and replace” using NotePad++ of the following:



NotePad++ was looking for a new line, line break, new line, line break and replacing it with just new line and line break. Simply put, it found blank lines in between good rows and data and removed them. This is something you could not do quickly by hand when you talking 300,000 lines of data.

The last part was using Excel because our data sets were now small enough that all the lines would fit into Excel. The next step was removing duplicates from our data. There is a quick way of doing this in Excel which uses the remove duplicate command under the data tool tab. You use it by selecting the whole data set and then remove duplicates must selected in the toolbar. Once is done, then you choose the column of data you want to reduce into unique results and excel does the rest.



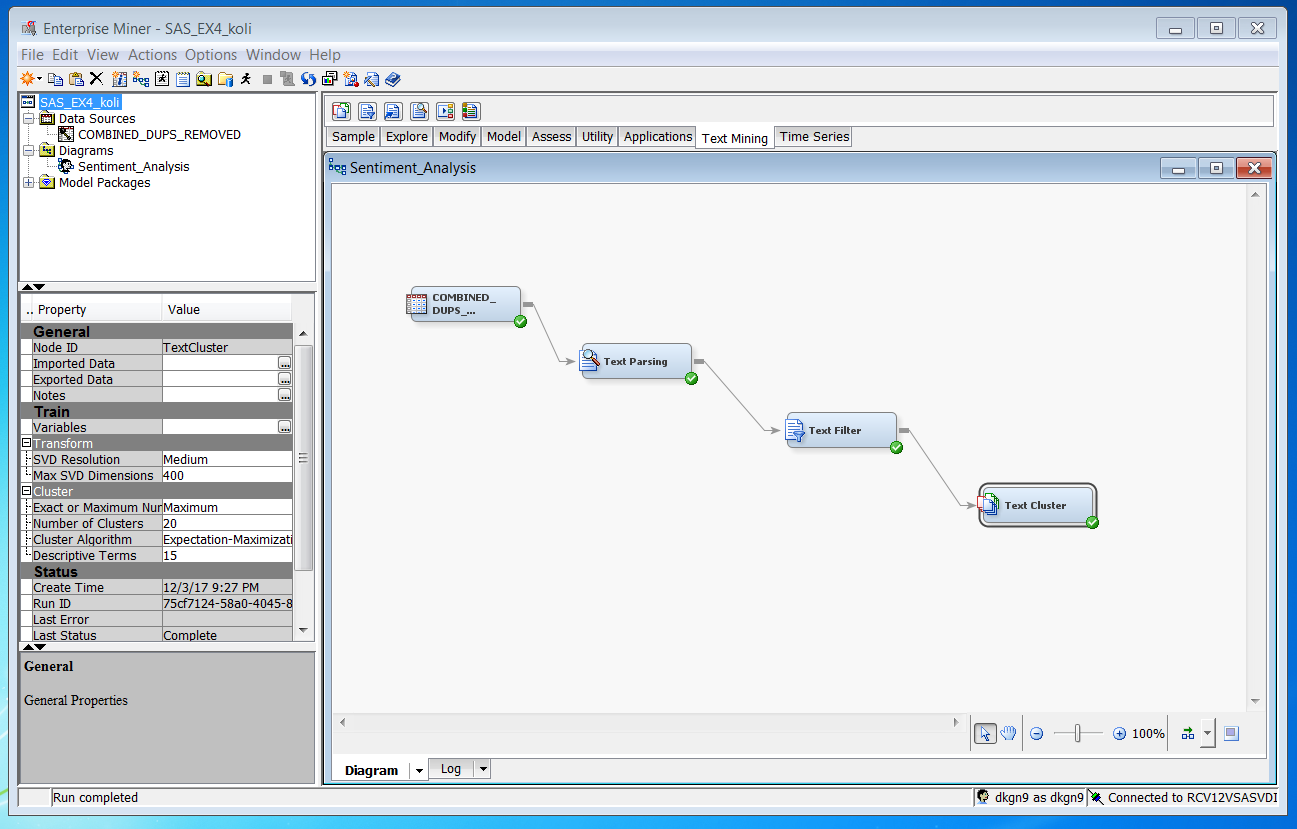
Overall, this way of cleaning up our data worked but it would not be the best way to do it if we keep collecting data. The best thing would be to write a Python program that would do all this work for us once we collected all the new data into its new folder. If we had a Python program like this, then it would be as simple as running the program to clean the data instead of running three manual steps. However, it did show us what we would need to build into our program and if the end results were what we needed. We did take it a step further and create SAS datasets out of our csv data sets. This was done by importing our csv datasets and saving them into a persistent SAS library where we could use them in SAS Enterprise Miner.

**SAS Enterprise Miner**

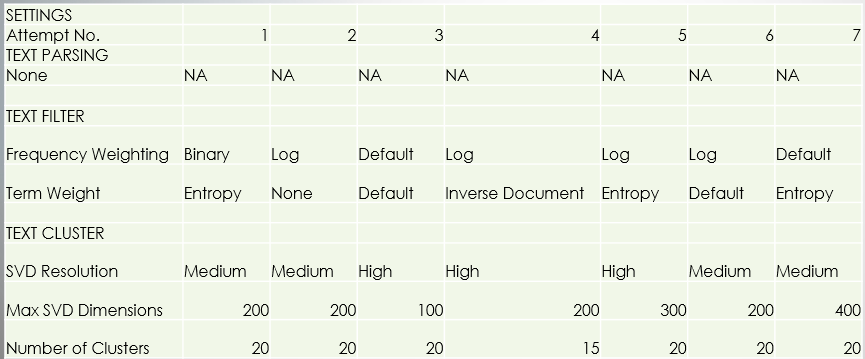
Data mining is a process of analyzing a large amount of data and identifying patterns establishing relationships between various entities to extract meaningful information. Text mining is a kind of data mining where data in the form of texts are analyzed to gather information. SAS Enterprise Miner is a tool that is descriptive and predictive assisting in the data mining process to find patterns and relationships for a given set of data.

For our term project we are using SAS Enterprise Miner 12.1 to extract information from the Twitter data (300,000 tweets) that we have collected from Oct 29th, 2017 till Nov 19th, 2017 regarding our topic of Gun Laws and its various comments from the people.

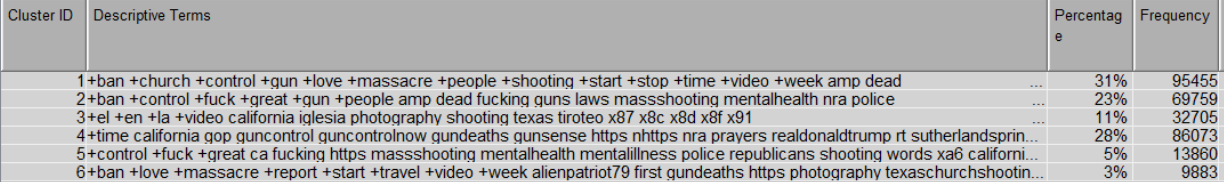
After collecting 300,000 tweets from Twitter we ran the data set (SAS file) in the tool using Text Parsing, Text Filter, and Text Cluster. The SAS E-Miner model was as follows:



We tried a variety of setting with parameters and got a variety of results. The clusters in each were a lot similar to each other but the number of clusters varied. Below are few of the combination of parameters that we tried:



After a number of runs, we settled for the below cluster:



A picture containing screenshot

Description generated with very high confidence

The reason for choosing this cluster are as follows:

-It gives a well-defined cluster based on the descriptive terms (e.g.. Gun control and NRA are in one cluster)

-The descriptive terms are close to what we were focusing our paper on. It includes terms like mental illness, GOP, NRA, etc.

Clusters and their result:

**Cluster 1-**



The Texas church shooting was an unfortunate event in American history. The first clusters focus on people tweeting regarding the shooting. Also, the caliber of the gun that was used for the shooting is being discussed. Amp is the term used to define the caliber of a gun. Prior to this attack, there have been 5 attacks in the U. S., people are angry about this and are insisting the government and the concerned authorities that it is the time that they stop talking about it and act on the crimes are happening around them.

**Cluster 2-**



The descriptive terms in Cluster 2 show the frustration of people towards the government and their lenient gun laws. The National Rifle Association (NRA) that advocates gun rights in the U. S. are also under a lot of scrutinies. People are of the opinion that the NRA is being heavily funded by the gun manufacturers resulting in the NRA supporting gun rights. Police are also being discussed in the conversation.

**Cluster 3-**



In U. S., Spanish is the second most spoken language. Hence, the cluster shows tweets in Spanish and people talking about the incident.

**Cluster 4-**



The government and its role in the gun laws is being discussed in the 4th Cluster. People, are of the opinion that others should have gun sense and it is high time that gun control should be made stringent. President Trump is also a part of this discussion.

**Cluster 5-**



People are pointing fingers towards the Republicans and blaming them for not so strict gun laws. The mental health and mental illness of a person is also being considered and few are blaming it for the mass shooting.

**Cluster 6-**



In this cluster, people are tweeting about the massacre. The “#love” is not only related to people showing concern for their loved ones but also commenting on the people who love guns and also the government and NRA who love the money that is contributed by gun manufacturers.

**CONCLUSIONS:**

After performing the analysis and looking at the results, we found that most percentage of people who have tweeted were against the current gun laws and expressed their concerns and frustrations on them. People were desperate that the government should start acting on changing these lenient gun laws. It is also found that people also blamed mental health or mental illness as one of the primary cause for such incidents in the United States. Overall, people are not happy with the current existing gun laws and are requesting the government to change them.

**FUTURE RECOMMENDATIONS:**

We plan to do more data cleansing by removing tweets that were generic i.e which were not related to the incidents, and then analyze the results. We also plan to collect more data if possible for doing the analysis. In addition, would be to create a full Python Script to do all data cleansing work to cut out manual work. We would also like to try other combinations of parameters and analyze the results.

**References:**

Getting started - Twitter Developers. (n.d.). Retrieved December 14, 2017, from https://developer.twitter.com/en/docs/basics/getting-started

KARAMBELKAR, B. (2015, January 05). How to use Twitter's Search REST API most effectively. Retrieved October 15, 2017, from https://www.karambelkar.info/2015/01/how-to-use-twitters-search- rest-api-most-effectively./

Nash, T. (2013, April 22). How to combine multiple CSV files into one using CMD -. Retrieved November 13, 2017, from http://www.tomnash.eu/how-to-combine-multiple-csv-files-into-one-using-cmd/

Tweepy Documentation. (n.d.). Retrieved October 07, 2017, from http://docs.tweepy.org/en/v3.5.0/index.html

**Appendix:**

1:

import tweepy

# Replace the API\_KEY and API\_SECRET with your application's key and secret.

auth = tweepy.AppAuthHandler('API\_KEY', 'API\_SECRET')

api = tweepy.API(auth, wait\_on\_rate\_limit=True,

   wait\_on\_rate\_limit\_notify=True)

if (not api):

    print ("Can't Authenticate")

    sys.exit(-1)

# Continue with rest of code

import sys

import jsonpickle

import os

import csv #Import csv

# Open/create a file to append data to

csvFile = open('result.csv', 'a')

#Use csv writer

csvWriter = csv.writer(csvFile)

searchQuery = 'Artificial Intelligence'  # this is what we're searching for

maxTweets = 10 # Some arbitrary large number

tweetsPerQry = 100  # this is the max the API permits

fName = 'tweets.txt' # We'll store the tweets in a text file.

# If results from a specific ID onwards are reqd, set since\_id to that ID.

# else default to no lower limit, go as far back as API allows

sinceId = None

# If results only below a specific ID are, set max\_id to that ID.

# else default to no upper limit, start from the most recent tweet matching the search query.

max\_id = 1

tweetCount = 0

print("Downloading max {0} tweets".format(maxTweets))

with open(fName, 'w') as f:

    while tweetCount < maxTweets:

        try:

            if (max\_id <= 0):

                if (not sinceId):

                    new\_tweets = api.search(q=searchQuery, count=tweetsPerQry)

                else:

                    new\_tweets = api.search(q=searchQuery, count=tweetsPerQry,

                                            since\_id=sinceId)

            else:

                if (not sinceId):

                    new\_tweets = api.search(q=searchQuery, count=tweetsPerQry,

                                            max\_id=str(max\_id - 1))

                else:

                    new\_tweets = api.search(q=searchQuery, count=tweetsPerQry,

                                            max\_id=str(max\_id - 1),

                                            since\_id=sinceId)

            if not new\_tweets:

                print("No more tweets found")

                break

            for tweet in new\_tweets:

                csvWriter.writerow([tweet.created\_at, tweet.text.encode('utf-8')])

                #f.write(jsonpickle.encode(tweet.\_json, unpicklable=False) +

                #        '\n')

            tweetCount += len(new\_tweets)

            print("Downloaded {0} tweets".format(tweetCount))

            max\_id = new\_tweets[-1].id

        except tweepy.TweepError as e:

            # Just exit if any error

            print("some error : " + str(e))

            break

print ("Downloaded {0} tweets, Saved to {1}".format(tweetCount, fName))

csvFile.close()