Paper 309-2012

Twitter and Facebook Analysis: It's Not Just for Marketing Anymore

Jodi Blomberg, SAS, Denver, CO

ABSTRACT

Think marketers have a hard time analyzing social media? Law enforcement has it even tougher. Crimes are constantly discussed all over Twitter and Facebook ---- and nobody's tagging them #crime. A vast multitude of other topics are also conversed about in these venues, making it no small task to capture intelligence from all of this text. We address two social media analytics applications for law enforcement. First, we show how to search social media sites for a specific set of people, gather up the publicly available data, and present it in a digestable form to the analyst. Second, we show how tracking events on Twitter can help us understand precursor activity to events such as riots. This will be useful to anyone using SAS to analyze social media.

INTRODUCTION

Recently, companies have discovered that social media analytics is crucial, especially for customer feedback and building goodwill. The analytics allow marketers to identify sentiment and detect trends in order to better accommodate the customer. There have been significant examples where companies, such as the airline industry, have used such analytical tools to reach customers based on feedback received.

Marketers aren't the only ones thinking about social media analytics. Despite the difficulties in analyzing social media, law enforcement has realized there is a wealth of free public information floating around on social media sites with the potential to aid in investigation and crime prevention. While companies are mainly interested in what social media has to say about their brand and their products, law enforcement agencies have two challenges in collecting and analyzing social media data that are unique to their goals. First, they are not always sure what subjects and key words they are looking for. Much of the intelligence in the data must be developed up front in order to determine what to search for in say, the Twitter stream. Second, they are restricted to "publicly" available social media data. They must be able to gather the data anonymously and without authentication tokens that require user approval.

In this paper, we will outline the concept and execution of two social media analytics applications that use SAS to address law enforcement issues. The applications incorporate social media in very different ways. The first is as an investigative tool to find social media related to specific people. Using an adaptation of our Social Network Analysis (SNA), we present Facebook and Twitter searches of multiple suspects in an easily digestable form for the analyst. The second application focuses on monitoring social media across a much broader spectrum, looking for the proverbial "needle in a haystack". In this example, we show how to collect and analyze historical Twitter data to try to understand precursors to dangerous activity at events, such as riots at concerts or flash mobs.

SOCIAL MEDIA RETRIEVAL

For crime investigation, we want to track social media for specific groups of individuals. While users can easily search for a single suspect's Facebook page or tweets, there is also an interest in searching for a group of individuals and viewing their social media and connections in a single format.

The goal of the application is to pull publicly available social media into a single interface. Rather than keep track of the multiple Twitter feeds and Facebook pages of a suspect, users can attempt to "follow" the story in a single interface. In order to facilitate this, we use the SAS SNA Server in an original way. The SNA Server is very adept at handling networked data. As such, it was the logical choice as the interface. Although SAS SNA was originally designed for fraud, the flexible interface allows us to customize it to our needs. In the following sections, we will outline the steps to enable this type of application.

There are four main steps to build our Social Media application using SAS SNA.

1. User entry of data

While the persons of interest are known, their Facebook IDs and/or Twitter handles likely do not exist in a database, so we must give users a means to enter that data.

2. Discover relationships

Collect data from the Facebook and Twitter APIs to determine if the suspects entered are related via social media (i.e. Facebook friends, Twitter followers)

3. Visualize the network

Render the relationships in a network diagram using SAS SNA.

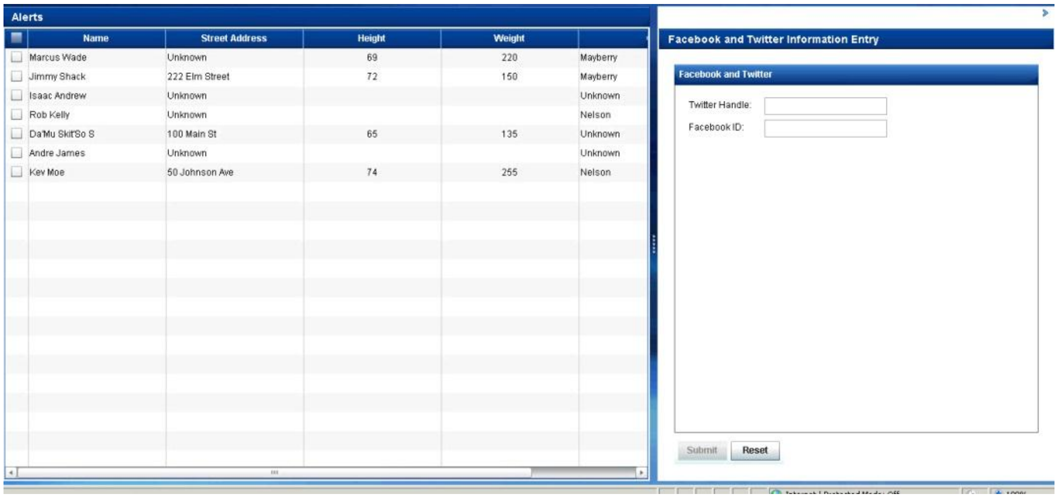
4. Gather Social Media

Collect data from the Facebook and Twitter APIs to accumulate actual tweets and Facebook posts related to these user IDs.

The following information relays one way SAS can access social media for this purpose, so a few caveats are in order. First, all information gathered is done so "anonymously", that is - without a user approved authentication token for Twitter or Facebook. This means we are restricted to viewing only publicly available data. Users who have protected their Facebook statuses to friends only, for example, will not be linked to anyone. Second, while sample code is included where appropriate, this is unlikely to suffice for gathering social media generally for businesses in a production level type capacity. Both Twitter and Facebook have user limits for accessing data and building an application that can exceed those limits requires additional registration with the companies themselves.

USER DATA ENTRY

Although a group of people of interest is known at the beginning of an investigation, their Facebook IDs and Twitter handles are rarely, if ever, stored in a database; and there is no automated way to differentiate between all of the Mike Smiths on Facebook and the "Mike Smith" of interest. Consequently, we need to provide a way to let users confirm and enter data about Facebook IDs and Twitter handles. SAS SNA provides an "alert disposition" panel so that users can select an entity from a list in the initial screen and indicate an action for that entity. In this SNA application, our entities are people of interest and our action is to save the Twitter Handle and Facebook ID to a dataset for use later. Since the alert disposition form is completely configurable via XML, we can use it as a place for the user to enter the person's Twitter handle and Facebook ID. A screenshot of this entry screen is shown in Display 1.



□	Name	Street Address	Height	Weight	
□	Marcus Wade	Unknown	69	220	Mayberry
□	Jimmy Shack	222 Elm Street	72	150	Mayberry
□	Isaac Andrew	Unknown			Unknown
□	Rob Kelly	Unknown			Nelson
□	Da'Mu Skit'So S	100 Main St	65	135	Unknown
□	Andre James	Unknown			Unknown
□	Kev Moe	50 Johnson Ave	74	255	Nelson

Twitter Handle:

Facebook ID:

Display 1: A Person is Selected On The Left Panel and Their Social Media Information Entered and Submitted On The Right Panel.

Finding Relationships

Now that we have the user IDs related to Facebook and Twitter, we want to determine if and how the persons of interest interact using social media. In order to determine who in the group of people of interest is connected by Facebook and/or Twitter, the application must call the APIs of the social media sites directly.

In Twitter, we can use the GET followers/ids command provided by the Twitter API to determine all of the followers of a Twitter handle. For example, the following will return an XML file with all of the users following the user named "twitterapi"

```
https://api.twitter.com/1/followers/ids.xml?cursor=-1&screen_name=twitterapi
```

An XML file with a list of user IDs following this screen_name is returned from this HTTP call. To match up the numeric IDs to our other suspects, we need to determine the Twitter screen names corresponding to those numeric IDs. We use the Twitter API GET users/lookup resource to make that match. For example, the following returns an XML file with all of the publicly available information about a the user ID 6253282, including screen name.

```
https://api.twitter.com/1/users/lookup.json?user_id=6253282&include_entities=true
```

Inside Base SAS, we can make these type of HTTP calls using a URL filename statement such as the following.

```
filename twitsrch url
“https://api.twitter.com/1/users/lookup.xml?user_id=6253282&include_entities=true";
```

These XML files can be converted to SAS data using the SAS XML Mapper. The resulting SAS data set contains the screen names for each of our people of interest and their followers so we can determine who is “following” whom.

Determining Facebook connections is not as simple as determining Twitter followers. Facebook only allows searches of very limited information without an authentication token. A basic token is assigned to any Facebook user and can be used to search any publicly available information. This will still not include a “friends” list. However, we can search for wall posts and statuses for any known Facebook IDs as long as we have a basic authentication token and the Facebook user has declined to make this information private. This will allow us to see when people in our list have posted to each other's walls or posted status updates for a given time range.

Much like the Twitter API, Facebook allows you to search based on an ID. The basic format for sending HTTP requests to the Facebook API known as the GRAPH API is the following

```
https://graph.com/ID/followers/CONNECTION_TYPE
```

Where a CONNECTION_TYPE is a things like statuses, friends, and posts.

A basic authentication token is required to search for statuses and posts. This does not require the user to approve the search but it does require sending some information to Facebook and revealing the search to Facebook. For that we need to “logon” with a Facebook profile. One way to accomplish this while remaining relatively anonymous is to create a blank Facebook page and profile and use that logon for searches. (see http://zesty.ca/facebook/ for an example of this approach). Another way is to register an application through Facebook, which will give you a basic authentication token. This approach is outlined very well in the 2011 SAS Global Forum paper “Social Networking and SAS: Running PROCS on Your Facebook Friends”.

In either case, Facebook will return the data in JSON format, so the SAS XML mapper cannot be used to convert this to SAS datasets. In the example using a registered application, the JSON file is parsed into data that can be read into SAS within a .NET application. Converting JSON to SAS in a generalizable way probably warrants its own SAS Global Forum paper (and we would love to see it !). We have written some naïve code to translate JSON to SAS using an infile statement and if-then statements to parse it as one long string of text. The following code is specific to the JSON file being returned. An example of the code is below.

```
filename twit url “https://api.twitter.com/1/followers/ids.xml?cursor=-
1&screen_name=twitterapi”;
```

The filename statement makes an HTTP call and names the file twit.

```
DATA _null_;
 N=1;
   INFILE twit recfm= v lrecl=2048 truncover;
   INPUT line $2048.;
   FILE “c:\temp\twit.txt.” lrecl=2048;
   PUT _INFILE_ ;
RUN;
```

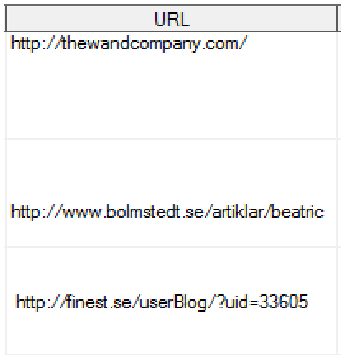
The data _null_ statement converts this JSON file to a text file called twit.txt.

```
DATA json(keep=colname colval row);
      length colname $ 1000;
      length colval $ 1000;
      retain state 1;
      retain row 0;
      INFILE twit firstobs=23;
      input ;
      len = lengthn(strip(_infile_));
      put len;
      line = strip(_infile_);
      put line;
      locu = index(line,"'url' =>");

      if (locU>0) then do;
             colname = "URL";
             line = substr(line, locu+8);
             locu2 = index("'",line);
             lineu2 = compress(substr(line, locu2+1),"'");
             colval = compress(substr(lineu2,1),",");
      end;

proc transpose data=json out=tout(drop=_NAME_ row);
      id colname;
      var colval;
      by row;
run;
```

This data JSON step reads in the text file as one long string and searches for the fields of interest. In this case, we capture the URL field and assign it to a variable called colval. We then transpose this file to get a dataset similar to the one show in Figure 2.

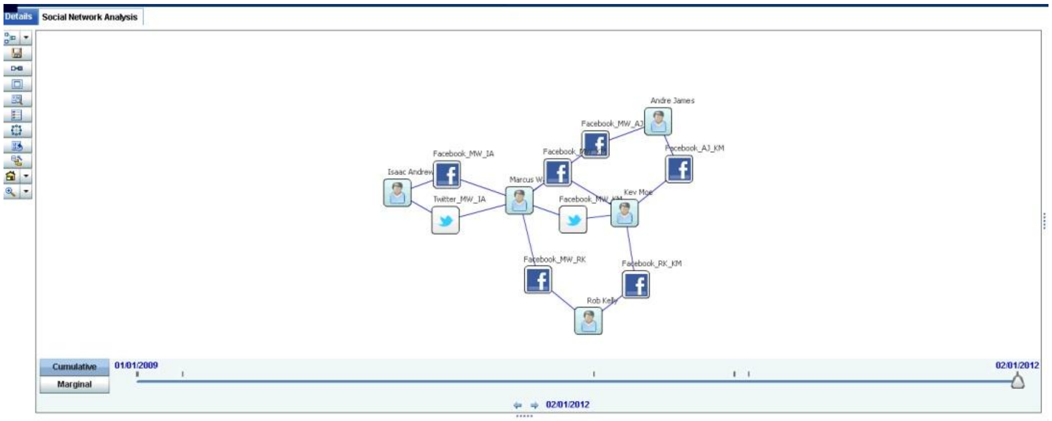


URL
http://thewandcompany.com/
http://www.bolmstedt.se/artiklar/beatric
http://finest.se/userBlog/?uid=33605

Figure 1: URLS read from a JSON response into a SAS dataset..

VISUALIZE THE NETWORK

Now that we have the Facebook and Twitter data, we can use this to create a network where links between people are based on Twitter and Facebook connections. In this application we consider two people to be connected on Twitter if one "follows" the other or vice versa. We consider two people to be connected on Facebook if one has posted to the other's walls or mentioned (e.g. "tags") the other person in a status update. In SAS SNA, this can be shown as in Display 2. Details on converting data into the nodes and links format necessary for display in SAS SNA is well outlined in the SAS Social Network Analysis Server 2.3: Administration Guide.



Display 2: Users Twitter and Facebook Connections as Displayed in SAS Social Network Analysis.

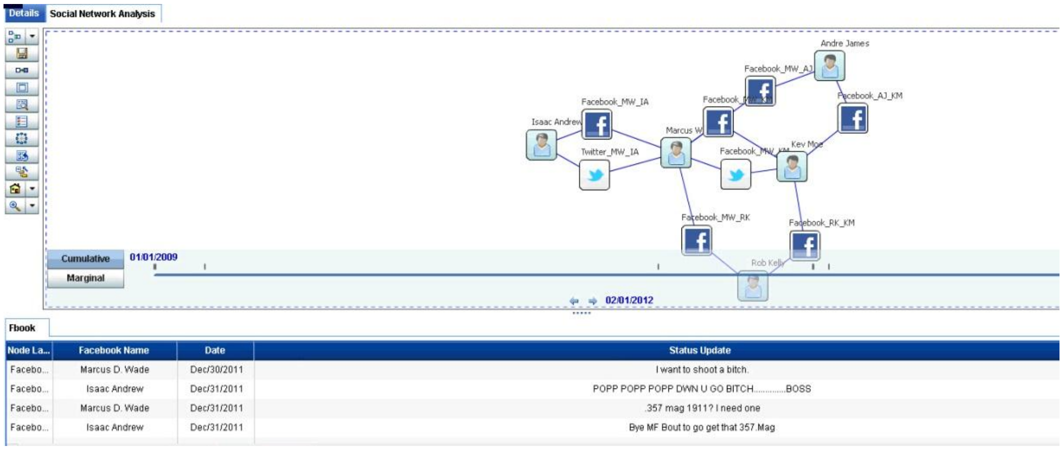
GATHER SOCIAL MEDIA DATA

SAS SNA allows users to see more detail for any node by clicking on that node to "Get Node Details" and developers can customize what is displayed. We go beyond just showing relationships between suspects on social media sites to actually displaying the person's Facebook status updates, walls posts and tweets. Here, we use "Get Node Details" to make another call to the Twitter and Facebook APIs to capture real time data.

The Twitter API allows a call to collect the last 200 tweets of each user, including retweets using the GET statuses request, shown as follows for user ID 1. However, please note that the Twitter API only allows you to search 5-7 days of history.

```
filename twit url
https://api.twitter.com/1/statuses/user_timeline.json?include_entities=true&include_rt
s=true&screen_name=twitterapi&count=200;
```

The Facebook Graph API allows a call to collect all wall posts and status updates using the same methods described in the previous section. The results are shown to the user are in Display 3.



Node La...	Facebook Name	Date	Status Update
Facebo...	Marcus D. Wade	Dec/30/2011	I want to shoot a bitch.
Facebo...	Isaac Andrew	Dec/31/2011	POPP POPP POPP DWN U GO BITCH.............BOSS
Facebo...	Marcus D. Wade	Dec/31/2011	.357 mag 1911? I need one
Facebo...	Isaac Andrew	Dec/31/2011	Bye MF Bout to go get that 357.Mag

Display 3: Facebook Status Updates as Displayed From Get Node Details in SAS Social Network Analysis.

Using this combination of Base SAS (to make HTTP calls and convert relationships, tweets, walls posts and statuses into SAS datasets) with the visual power of SAS SNA, results in an application which speeds investigation and creates a story line that can be crucial for investigators who want to harness the information available in social media. This information around conversations is rarely available anywhere else to investigators.

MONITOR SOCIAL MEDIA

Unlike their commercial counterparts who monitor the Twitter stream for any mention of a product, law enforcement clients don't necessarily know what they need to monitor on Twitter. Rather, text analytics are used to analyze past events to determine what should be monitored in the real time Twitter stream.

In order to analyze historical tweets, we need to do the following:

1. **Collect Historical Data.**

   Collect historical tweets for specific times and events of interest

2. **Analyze Historical Data.**

   Analyze the tweets for patterns and anomalies to track in the future.

3. **Monitor Real Time.**

   Monitor the real time tweet stream for items of interest.

COLLECT HISTORICAL DATA

To determine what might be of interest, we can learn from historical patterns of tweets around events of interest and try to learn patterns and keywords that occur commonly across these events. For example, we can analyze tweets in the 48 hours leading up a concert or sporting event where criminal behavior occurred and determine how the nature of the tweets leading up to the event differ from tweets leading up to a concert where no crimes occurred. As stated previously, the Twitter API only allows you to search 5-7 days of history, so a different method of data collection is required if we want to learn from events occurring in the last few years. Fortunately, the Topsy service (http.topsy.com) has archived tweets for the past three years and has an API that allows us to search for specific terms related to the time of specific events.

Ideally, we would restrict our historical data collection to tweets originating from a specific location. Using the Twitter API, you can collect the Geotagging related to a tweet. Not all users enable Geotagging, which assigns a latitude/longitude to the origin of the tweet. If users do not enable Geotagging, the latitude/longitude values will fall back to their Twitter profile. However, for historical data collection, we are dependent on the Topsy Otter API, which currently does not include location information of either type.

The Otter API (http://code.google.com/p/otterapi/) provides access to Topsy search results, URL information and author information. Again we can use a URL filename to make a simple HTTP call to the API to fetch tweets. Queries to Topsy can be defined by a query string of key words and a time range. Time ranges must be expressed in a UNIX timestamp format.

For example, the following query fetches all tweets mentioning "philly" from Thu, 28 Jul 2011 05:00:00 GMT (unix timestamp 1311829200) to Thu, 28 Jul 2011 07:00:00 GMT (unix timestamp 1311836400).

```
filename twit1 url
'http://otter.topsy.com/search.txt?q=philly&mintime=1311829200&maxtime=1311836400&perp
age=100&page=1'  recfm=s debug encoding = "utf-8";
```

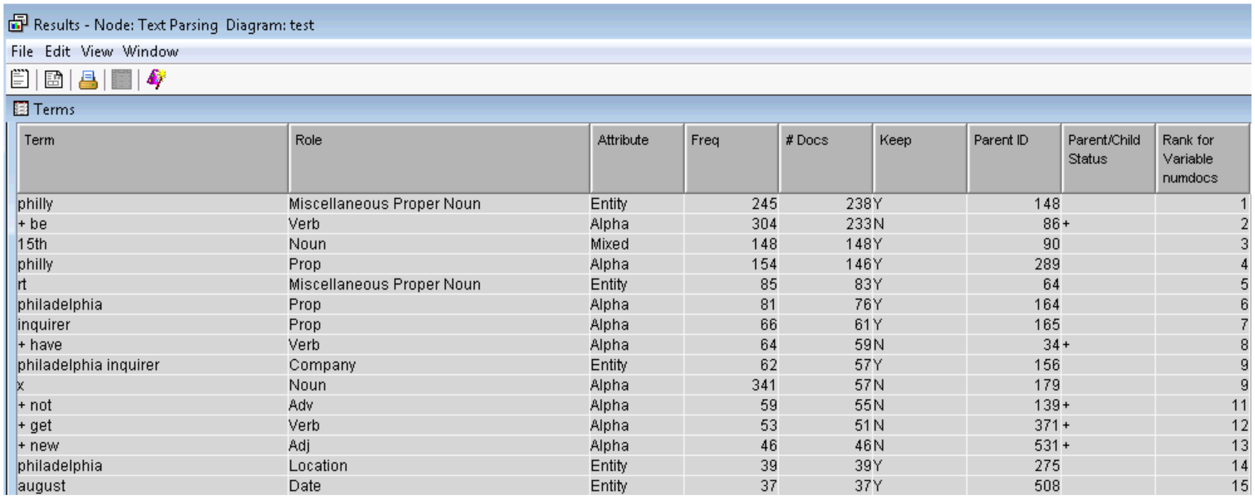
Like the Facebook Graph API, the Otter API returns JSON files and they must be parsed using SAS or by other means in order to be translated into SAS datasets. To collect for the 48 hours preceding an "event", we search for any or all keywords that may be related to an event of interest. As an example, there was an incident in Philadelphia where a group of 20-40 teenagers assaulted and robbed pedestrians and damaged property Using the description of the attacks and the locations of the attacks, we can attempt to gather all tweets that might be related to the attacks by creating a list of terms to search for and gather tweets on each keyword possibility. First, we might gather all tweets mentioning "philly" or "Philly" in the 48 hours previous to the attack and then create more queries about the event such as street names or parks where the attacks occurred. Second, we leverage the institutional knowledge of agency analysts, investigators and existing databases to build search terms; for some events, we have collected as many as 100 different search queries related to keywords. Defining the list of keywords to search is more art than science and relies heavily on descriptions of the attacks and domain knowledge.

Once these keyword files are collected, we simply append them, weed out duplicate tweets and they are ready for analysis in SAS Text Miner.

ANALYZE HISTORICAL DATA

To analyze the historical data, we use SAS Text Miner which can parse the tweets, provide word counts and extract entities.

Using the Text Parsing node of SAS Text Miner, we can quickly view all of the terms appearing in the tweets, the part of speech, how many times the term has occurred and in how many tweets the term appears. The Text Parsing node is not limited to merely tallying individual words but also phrases and entities, such as company names. An example of the Terms window describing all terms appearing in a set of tweets is show in Display 4.

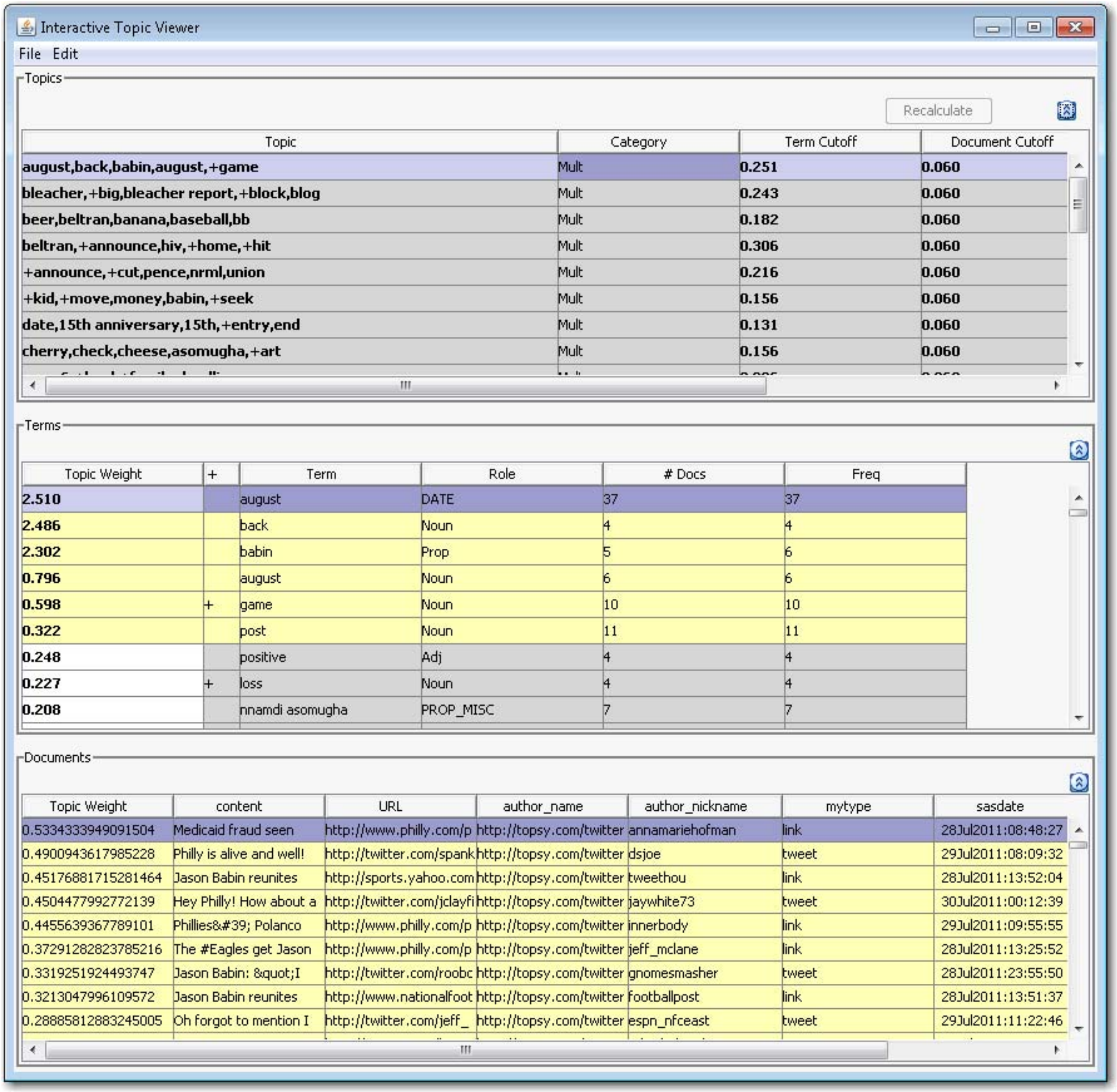


Term	Role	Attribute	Freq	# Docs	Keep	Parent ID	Parent/Child Status	Rank for Variable numdocs
philly	Miscellaneous Proper Noun	Entity	245	238	Y	148		1
+ be	Verb	Alpha	304	233	N	86	+	2
15th	Noun	Mixed	148	148	Y	90		3
philly	Prop	Alpha	154	146	Y	289		4
rt	Miscellaneous Proper Noun	Entity	85	83	Y	64		5
philadelphia	Prop	Alpha	81	76	Y	164		6
inquirer	Prop	Alpha	66	61	Y	165		7
+ have	Verb	Alpha	64	59	N	34	+	8
philadelphia inquirer	Company	Entity	62	57	Y	156		9
x	Noun	Alpha	341	57	N	179		9
+ not	Adv	Alpha	59	55	N	139	+	11
+ get	Verb	Alpha	53	51	N	371	+	12
+ new	Adj	Alpha	46	46	N	531	+	13
philadelphia	Location	Entity	39	39	Y	275		14
august	Date	Entity	37	37	Y	508		15

Display 4: Sample of Text Parsing Node Results from SAS Text Miner

Once the text is parsed, we pass it through the Text Miner node and get acquainted with our data using the interactive feature, where we can view the tweets themselves and look at tweets associated with certain phrases or words to help determine what is important and what is not. It is usually important in text mining projects to drop certain terms that will not help with analysis, such as small words like "is" or "and". Usually there are additional words determined by domain knowledge specific to a project that can also be dropped for lack of information.

In order to determine patterns or keywords specific to the tweets, we want to analyze tweets related to events of interest and some group of "non interesting" events like concerts where nothing happened. Then we can build a predictive model, such as a regression, to differentiate between the two groups of tweets. The inputs to this model are not the tweets themselves, but keywords, phrases, topics, and/or other output from the Text Miner nodes. Depending on the events and the nature of the tweets, we may use the Text Topics node to try to group the tweets into topics. An example of the interactive Topic Viewer on some of the tweets from our example is shown in Display 5.



Topic	Category	Term Cutoff	Document Cutoff
august,back,babin,august,+game	Mult	0.251	0.060
bleacher,+big,bleacher report,+block,blog	Mult	0.243	0.060
beer,beltran,banana,baseball,bb	Mult	0.182	0.060
beltran,+announce,hiv,+home,+hit	Mult	0.306	0.060
+announce,+cut,pence,nrml,union	Mult	0.216	0.060
+kid,+move,money,babin,+seek	Mult	0.156	0.060
date,15th anniversary,15th,+entry,end	Mult	0.131	0.060
cherry,check,cheese,asomugha,+art	Mult	0.156	0.060

Topic Weight	+	Term	Role	# Docs	Freq
2.510		august	DATE	37	37
2.486		back	Noun	4	4
2.302		babin	Prop	5	6
0.796		august	Noun	6	6
0.598	+	game	Noun	10	10
0.322		post	Noun	11	11
0.248		positive	Adj	4	4
0.227	+	loss	Noun	4	4
0.208		nnamdi asomugha	PROP_MISC	7	7

Topic Weight	content	URL	author_name	author_nickname	mytype	sasdate
0.5334333949091504	Medicaid fraud seen	http://www.philly.com/p	http://topsy.com/twitter	annamariehofman	link	28Jul2011:08:48:27
0.4900943617985228	Philly is alive and well!	http://twitter.com/spank	http://topsy.com/twitter	dsjoe	tweet	29Jul2011:08:09:32
0.45176881715281464	Jason Babin reunites	http://sports.yahoo.com	http://topsy.com/twitter	tweethou	link	28Jul2011:13:52:04
0.4504477992772139	Hey Philly! How about a	http://twitter.com/jclayfi	http://topsy.com/twitter	jaywhite73	tweet	30Jul2011:00:12:39
0.4455639367789101	Phillies&#39; Polanco	http://www.philly.com/p	http://topsy.com/twitter	innerbody	link	29Jul2011:09:55:55
0.37291282823785216	The #Eagles get Jason	http://www.philly.com/p	http://topsy.com/twitter	jeff_mclane	link	28Jul2011:13:25:52
0.3319251924493747	Jason Babin: &quot;I	http://twitter.com/roobc	http://topsy.com/twitter	gnomesmasher	tweet	28Jul2011:23:55:50
0.3213047996109572	Jason Babin reunites	http://www.nationalfoot	http://topsy.com/twitter	footballpost	link	28Jul2011:13:51:37
0.28885812883245005	Oh forgot to mention I	http://twitter.com/jeff_	http://topsy.com/twitter	espn_nfceast	tweet	29Jul2011:11:22:46

Display 5: Text Topics of Sample Tweets

Based on topics, keywords and/or entities, the predictive model can determine what elements of the tweets are important in differentiating between events with negative vs positive outcomes. These identified elements become the inputs for us to monitor the real time Twitter stream.

MONTIOR REAL TIME

Once we have determined what we are looking for, we can now monitor the real time Twitter stream to filter only those tweets that interest us. Twitter has a separate API for real time streaming Twitter monitoring that allows developers to pull public statuses from all users, filtered in various ways such as userid, keyword, and geographic location.

For law enforcement, narrowing the Twitter stream to tweets of interest involves implementing two types of filters: geographic and key words. Collecting only tweets from a jurisdiction does create some potential room for "missing" information but is likely to reduce the amount of data collected to a reasonable size. Filtering to only tweets in English is necessary whenever our historical data analysis is based on English tweets, which is most likely the case.

Last but hardly least, we search for certain key words or patterns of interest. Twitter does restrict access to the Twitter stream to 400 track keywords, 5,000 follow userids and 25 0.1-360 degree location boxes.

Twitter Streaming API queries are not much more complicated to write than Search API queries. However, implementation of a system that monitors Twitter is quite a bit more complicated. Data volumes can quickly increase and Twitter encourages developers to plan for traffic to double every few months. In addition to storage issues, developers should consider how often to "catch" the stream; this decision will vary based on the nature of the request. The more specific the search, the less data it returns, and less often it can be "refreshed" without missing something.

CONCLUSION

This paper outlined two ways in which law enforcement clients can harness the information contained in social media sites such as Facebook and Twitter. These relatively simple examples are intended to be the basis for thinking about how the enormous amounts of social media data can be collected and analyzed to turn tweets and posts into useful information. The amount of data in this arena grows daily and shows no sign of slowdown. New methods, ideas, and models will have to continue to evolve to analyze it.

REFERENCES

Hemedinger, Chris and Slaughter, Susan, 2011. "Social Networking and SAS: Running PROCS on Your Facebook Friends", *Proceedings of the SAS Global 2011 Conference*. Available at http://support.sas.com/resources/papers/proceedings11/315-2011.pdf .

Daily News Staff Report 31 July, 2011 http://articles.philly.com/2011-07-30/news/29833311_1_locust-teen-streets

ACKNOWLEDGMENTS

The Facebook example application uses the Facebook C# SDK to gather data using the Facebook API; the SDK is available from http://Facebooksdk.codeplex.com. It uses the Json.NET library to parse the JSON-formatted responses. Json.NET is available from http://json.codeplex.com/. Information regarding the Twitter REST API for search is located at: https://dev.twitter.com/docs/api. For real time Twitter searches, the Twitter Streaming API is found at https://dev.twitter.com/docs/streaming-api. The Topsy Otter API is documented at http://code.google.com/p/otterapi/

RECOMMENDED READING

For more information about how to use the XML LIBNAME engine and SAS XML Mapper, see *SAS9.2 XML LIBNAME Engine: User's Guide***.** It is available online at:
http://support.sas.com/documentation/cdl/en/engxml/62845/HTML/default/viewer.htm
For more information about what you can do with SAS Social Newtork Analysis, see the SAS product page online at:
http://support.sas.com/software/products/sna/index.html.
For more information about the JSON standard, see http://json.org .

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jodi Blomberg
SAS
6501 S Fiddlers Green Circle
Greenwood Village, CO 80111
919-531-9778
Jodi.blomberg@sas.com