Extraction of Semantic Activities from Twitter Data

Aleksey Panasyuk
Air Force Research Lab
Rome, NY, 13441
Aleksey.panasyuk@us.af.mil

Erik Blasch
Air Force Research Lab
Rome, NY, 13441
Erik.blasch.1@us.af.mil

Sue E. Kase
Army Research Lab
APG, MD
Sue.e.kase.civ@mail.mil

Liz Bowman
Army Research Lab
APG, MD
Elizabeth.k.bowman.civ@mail.mil

Abstract— With the growing popularity of Twitter, numerous issues surround the usefulness of the technology for intelligence, defense, and security. For security, Twitter provides a real-time opportunity to determine unrest and discontent. For defense, twitter can be a source of open-source intelligence (INT) information related to areas of contested environments. However, the semantic content, location of tweets, and richness of the information requires big data analysis for understanding the use of the information for intelligence. In this paper, we describe some results in using twitter data to determine events, the semantic implications of the results from the data, as well as discuss pragmatic uses of twitter data for multi-INT data fusion. The results collected during the period of Egypt Arab spring conclude that (1) many tweets are clutter or noise in analysis, (2) location information does not always convey the accuracy of the information, and (3) the aggregate processing of the twitter data results in real-time trends of possible events that warrant more conventional information gathering.

I. INTRODUCTION

Over the last decade, there has been a surge of the use of constrained messages sizes in 140 characters or less, known as "tweets." The popularity of tweets has three emerging issues: (1) big data as the number of users grows, (2) semantic extraction of meaningful content from cryptic phrases and non-standard terminology, and (3) the large amount of semantic clutter that reduces the signal-to-noise ratio of identifying salient content (e.g., key words of phrases).

A. Twitter as a source of Intelligence

While the use of open source information becomes popular such as Facebook, imagery, and text; it is well established that tweets are being used by anyone anywhere from distributed mobile platforms. The presentation of different semantic formats and the number of users require pragmatic approaches to searching and deriving meaningful content from tweets. Meaningful content is further exacerbated as the source of tweets does not always correspond to the location of the user; however, timing and general trends over many users can determine the status of an emerging event.

Twitter data, while popular, suffers from various content issues that have to be solved with advanced and tailored methods. Examples of problems include users with hidden meanings, masked source of origin, possible deceit and deception, as well as non-descriptive and non-important discussions of social issues (e.g., where to go for dinner). However, tweets do provide a forum where users can express their social and political views, news reports of immediate actions that are not available to the regular media, and links of semantic content such as to video collected and posted to the web from cell phones.

It is the interplay between the availability and enormity of tweets to that of extraction of meaningful content that is derived from users close to the action. Tweets provide reports that are not available from other intelligence sources of information.

B. Twitter Semantic Extraction

An approach taken by most search engines over twitter data is to organize documents and their terms in a Vector Space Model (VSM) [1]. A Vector Space Model is a two dimensional array. The rows of this array are a list of terms from all documents that a user is searching through. The columns are the names of documents. The VSM ranks all terms using frequency analysis utilizing the bag of words hypothesis. Bag of words hypothesis states that two documents tend to be similar if they have an equivalent distribution of analogous words [2]. In this way, a search engine query can be seen as a vector of terms which can be used with a VSM in order to find documents that are closest to this vector via some distance measure.

With successful implementation of VSM by the search engines, researchers have attempted to apply VSMs to other areas of natural language processing (NLP). For a long time it had been considered that to understand the meaning of words it is enough to consider statistical word usage, the so called statistical sematic hypothesis [3,4]. The benefit of VSMs is that they easily consume large amounts of data and require far less labor than other approaches [5]. For example, Rapp [6] developed a vector representation of word meanings mainly from British National Corpus. The British National Corpus is not a lexicon but is simply a text corpus containing 100 million words annotated with parts of speech (www.natcorp.ox.ac.uk/). Rapp's VSM was used on multiple choice questions from Test of English as a Foreign Language (TOEFL) achieving a score of 92.5% where the average human score was 64.5% [6]. TOEFL is a well-structured text making preprocessing and identification of terms an easy task.

However, Twitter is more complicated as the text is unstructured.

Twitter has a lot of informal and abbreviated text. For example, current tools, while practical on news articles and similar types of well written documents, perform quite poorly for Part-of-Speech (POS) tagging and Named Entity Recognition (NER) when applied to tweets. The accuracy of tools falls from 97% accuracy for news articles to about 80% for tweets [7]. In [8], Finin *et al*. experiment with crowdsourcing for POS tagging on tweets. Crowdsourcing is made available by a service such as Amazon's Mechanical Turk which allows for tasking and collecting results from a "crowd" of people that are willing to do the work by hand. Others [9, 10] propose lexical normalization of tweets which may be useful as a preprocessing step for the upstream tasks like POS tagging. In [11], Gouws *at al*. try to properly tag parts of Twitter speech. They mention that it is hard to tag words within Twitter because of the conversational nature of text, lack of conventional orthography, and limit of 140 characters. Messages on Twitter are filled with grammatical errors, abbreviations, slang, words in another language, and URLs. The authors take a number of steps to give them an advantage such as using the Metaphone algorithm [12] in order to remove alternative spelling for words, developed expression-style rules for capturing known structures like URLs, keeping capitalized words that follow an expected distribution, as well as using known lexicons such as WordNet in conjunction with their algorithms, etc. In trying to understand statistical semantics in Twitter, the authors use unsupervised word representations as extra word features. 1.9 million features from 134,000 unlabeled tweets are used to construct these distributional features via an approach outlined in [13]. Even though the training set is limited to 1000 records (tweets), the unsupervised word representations capture enough content to achieve nearly 90% accuracy on the 500 testing records.

To perform further analysis such as sentiment analysis, it would make sense to perform all of the steps in the papers cited i.e. set of tweets needs to be found, the features from tweets are extracted, preprocessed, tagged, and statistical analysis is performed. Sentiment analysis can be used in order to identify anger, tension, and other emotions that may be tied to significant offline events [14, 15]. Tweets can even be used for predicting events like earthquakes [16] and box office sales [17]. Twitter has a lot of data, about 400 million tweets per day, which is beyond what human beings can handle even with crowdsourcing. Finding relevant text is becoming increasingly challenging such that there is a growing need for automatic text understanding that scales to the Web. There are systems known such as open Information Extraction (IE) systems that are being developed to address text understanding [18, 19, 20], but in order to use such systems we need to know the features of interest.

The rest of the paper is as follows. In Section II, we describe the use case of Twitter data from the Egyptian uprising. In Section III, we discuss how the data is structured. Section IV and VI describe analytics and visualization, respectively. Finally, Section VII describes conclusions.

II. TWEETS FROM THE ARAB SPRING SCENARIO

The Egyptian uprising of 2011 is an example of an important historical event which has been captured via social media sites such as Facebook and Twitter. Some argue that without social media, like Twitter, the uprising would have not achieved the same level of success [21, 22]. Social media allowed countless participants to be involved. Twitter has become an important social media site since its inception in 2006. It is a micro blogging service which allows users to post messages up to 140 characters in length. Once a message is posted, any twitter user in the world can see it, repost it, and reply to it. A user may search for messages based on topic or person of interest. A user may choose to "follow" another user which will cause all of the messages posted by a user that is being followed to be displayed on that user's Twitter timeline.

In regards to Egypt, January 25th 2011 had become known as the "day of rage" with protests in Cairo. Social media and Internet played such a key role that the Egyptian government had begun limiting Internet access on January 27th [23]. Egypt related topics continued circulation until February 11 when President Mubarak resigned. During the course of events, it was noted that information was coming from tweets, but the intelligence sources were not mobilized to use the technology and even if available, to what extent that content could be gleamed from the experience of multiple users presenting tweets.

In this paper we attempt to analyze 738,717 tweets from that time period in 2011. The Egypt Twitter data has been grouped into 10 classes by Army Research Lab (ARL) [24]. The 10 classes come from ranking the Twitter data using the Tri-HITS model described in [25]. Tri-HITS paper describes an algorithm for ranking tweets not only based on the textual content within the tweet but by also considering the referenced web documents and popularity of users. The results of the Tri-HITS model show improvement over popular algorithms such at TextRank [26]. Once the Egypt tweets are ranked they are equally broken down into 10 groups with first group being the tweets that had a ranking that beat 90% of other tweets, second group being the tweets that beat 80% of tweets but are not part of first group, and so on. Given these initial classes we investigated whether the features behind those classes made sense and could be used for deeper searches by the analyst. Our interests for this data had been whether we could use it as a source for proactive situational analysis.

In relation to the above methods, our novel approach is to apply a VSM model to the groupings made by the Tri-HITS model in order to extract top one hundred features associated with each grouping. The top features that are extracted can be used for evaluating the quality of grouping and can be used by the analyst when searching for similar events. The algorithm is fast and straight forward to implement and does not require human in the loop involvement. To the best of our knowledge we are not aware of papers applying a VSM model on rankings generated with Tri-HTTS model for the Egypt data.

III. STRUCTURING DATA

The first task was to properly structure the data within a MySQL database. This is the environment that we have been using:

- Windows 7 64 bit
- Eclipse 4.2.0 with Python IDE plugin (PyDev)
- Python 2.7.x with MySQL-python connector
- MySQL 5.5 with MySQL Workbench 5.2 CE

Each tweet is limited to 140 characters and is associated with some class label. Figure 1 shows the basic tweet to class label structure. There are a total of 10 classes c_0, c_1, …, c_9.

```
1 • SELECT * FROM egyptuprising2.tablecontainingrawdata;
```

Filter: Export: Autosize: Fetch rows:

class	text
c9	@Panoptique Is Stephen Cohen nuts? Deaf? Blind? Stupid? Related to Sarah Palin?
c9	#Imagine what would happen if #censorship occurred in the United States? #foodforthought #Egypt #Jan25 #change
c9	Egypt.
c9	@djeratic Egypt?
c9	#Egypt Ah! Now I know why the military is in Sharam El Sheik! With Israeli agreement! Hosni Mubarak is there! Not Cairo!
c9	Proof I miss being in Seattle.. Watching the news on Cairo I keep thinking... What's wrong with channel 7! If only they still had JP Patches
c9	Helicopters are still roaming the air, protesters are still in Tahrir. I hear them chanting still
c9	so what to pack ....
c9	Morning all- gym- done bring on the day
c9	For those interested, this is the #Google search I use to get these messages from #Egypt http://bit.ly/g5l4pZ
c9	What will happen today? #Egypt
c9	'Good morning tweeps have a super day \u263a
c9	'La \u201cMarcha del mill\xf3n\u201d parte desde Tahrir y va hasta el Palacio Presidencial, inicia a las 9:00 hora de #Egipto 7:00 GMT
c9	#Egyptians #Cairo #Mubarak
c9	Another sweaty day Ethekwini. Love it, wouldn't have it any other way!

Figure 1. Tweet to Class Data

Figure 2 shows a vision for all of the main steps listed in the referenced papers. The first step is to *find information of interest* whether it is related to some event, an organization, a product, etc. Open IE systems can help retrieve the data we are interested in, if we have a broad enough set of terms that cover the topic of interest. The event of interest for us is the Egypt uprising data supplied by the ARL [24]. We don't use an IE system in this paper, but it would be the goal to use the terms extracted from this research with an IE system in the future in order to find related events.

The second step is to *tokenize*, i.e. extract features. Most often terms of interest are separated by white space, but researchers need to consider how they want to treat URLs, punctuation, and multiword features such as "daylight savings". We had two approaches. Our first approach had been to simply use white space as delimiter, join on punctuation, and disregard features over 50 characters in length (this gets rid of most websites). Our second approach was to focus on specific topics and people on Twitter:

Tokenize Approach A. Feature = anything that is no more than 50 characters in length and that contains only digits and ascii_letters i.e. any other characters are removed.

There were a total of 782,713 features using this approach.

For example:

http://www.google.com is converted to → httpwwwgooglecom which becomes our feature.

Tokenize Approach B. Feature = Twitter hashtags (Twitter topic that begins with "#") and Twitter at-mentions (at-mentions begin with "@").There were a total of 106,322 features using this approach.

For example:

#egypt and @youtube would be the structure of our features.

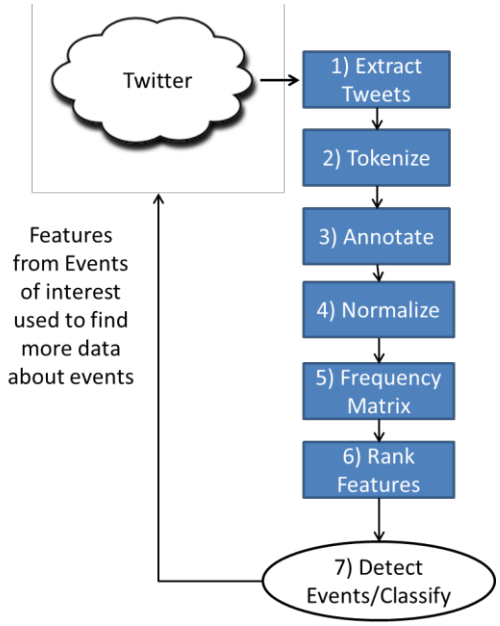

Figure 2. Main Steps for processing

The third step is to *normalize*. Normalizing reduces similar features. For instance it is common to use a stemmer in order to turn words like fixing, fixed, fixer → fix. The Metaphone algorithm mentioned in [11] will map words from a set like {thangs thanks thanksss thanx thinks thnx} to a single key, but sometimes this is not desirable as {war we're wear were where worry} are also mapped to a single key. A researcher may also choose to remove common stop words like "the". Normalizing will typically increase recall (when system identifies a relevant tweet as relevant), but decrease precision (when a tweet that is identified as relevant is truly relevant). In this paper, the only normalization we do is to turn everything to lowercase and consider only printable characters.

The fourth step is to perform *annotation*, for example roll can be tagged as a verb roll/VB (to rotate around an axis) or

a noun roll/NN (a small loaf of bread). Annotation is the inverse of normalization so it tends to improve precision and decrease recall. Annotation can be performed using well established lexicons that contain basic rules of grammar for such operations; examples include WSJ and Brown corpora as well as WordNet and Moby lexicons. In this paper we have not attempted any annotation.

The fifth step is to use the features in a *frequency matrix*. It begins by recording how many times each feature appears in each class. Table 1 is the result of an SQL query which orders features by total times used over all classes using tokenizing approach A.

feature	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	Total Times Used
egypt	35315	33624	39010	37230	32318	18043	16740	18092	18481	20441	269294
the	13506	14938	23251	22736	18969	12753	15956	17324	17224	17970	174627
in	19952	19225	23614	21073	17409	10044	12205	13437	13353	14429	164741
to	12106	11507	16759	17178	14557	10164	13285	14364	14575	14998	139493
tahrir	21220	19984	21232	13619	7300	2323	7745	8813	9148	10007	121391
cairo	21213	15749	17590	16029	10397	3181	5868	6736	6780	7648	111191
25-Jan	15708	14561	16954	13976	9218	5093	7094	7878	7808	8432	106722
of	9882	10240	15503	14815	11854	7027	8054	8704	8617	9126	103822
a	6527	6706	10892	12640	11204	8137	10694	11381	11517	11567	101265
…	…	…	…	…	…	…	…	…	…	…	…

Table 1: Feature Counts for each Class

	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9
Classical Accuracy	0.19078	0.141639	0.158196	0.144157	0.093506	0.028608	0.052774	0.06058	0.060976	0.068783
Percent Contribution	0.191419	0.141558	0.157702	0.143607	0.093232	0.029631	0.053232	0.060323	0.060845	0.068451
TF-IDF	0.252498	0.1849	0.206659	0.187659	0.120601	0.037859	0.068309	0.07751	0.078192	0.088101

Table 2: Example Feature Scores

The sixth step is to *rank features*. The counts can then be used to calculate probabilities and to rank features. Classical frequency only considers how probable a term is to occur within a class. For example consider that the feature "to" was seen 1000 times for class 1 and 2000 times for class 2. Classic accuracy is going to be: "to" appears 1000/3000 = 33.33% of time in class 1, and 2000/3000 = 66.66% of time for class 2.

Classic accuracy could be taken to mean that the feature "to" is associated with class 2 with 66.66% accuracy. But consider that class 1 had a total of 5000 records and class 2 had a total of 10000 records. This additional information tells us that "to" had appeared in every record of class 1 and every record of class 2. Hence "to" is not a relevant feature.

Instead of using classical frequency, most papers use the term *frequency multiplied by inverse document frequency* (TF-IDF) [27]. In this way the greatest ranking is when the feature is frequent for a particular class and not frequent in all other classes, calculated as:

$$\text{TF} - \text{IDF for feature i in class j} = \left(\frac{\text{Total times feature i appeared for class j}}{\text{Total number of tweets in class j}}\right) * log\left(\frac{\text{Total number of classes}}{\text{Total number of classes that have feature i}}\right)$$

Besides using the TF-IDF measure we rank features by calculating how each feature contributes to each class using the following percent contribution formula:

$$\text{Feature i contribution to class j} = \frac{\left(\frac{\text{Total times feature i appeared for class j}}{\text{Total number of tweets in class j}}\right)}{\sum_{k=0}^{n-1}\left(\frac{\text{Total times feature appeared for class N}}{\text{Total number of tweets in class N}}\right)}$$

(where n is the number of classes involved)

Both measures consider the number of records in all classes which is better than using classical frequency. Using the percent contribution formula on the example from above, we see that class 1 and class 2 evenly contribute to this ratio, i.e.:

denominator = 1000/5000 + 2000/10000 = 2000/5000

%contribution by "to" feature from class 1 =
(1000/5000)/(denominator) = 50%
%contribution by "to" feature from class 2 =
(2000/10000)/(denominator) = 50%

TF-IDF will rank these two classes as equals as well:

(1000/5000)*(log(2/2)) = 0
(2000/10000)*(log(2/2)) = 0

Table 1 is used to generate measures for each feature using classical accuracy, percent contribution, and TF-IDF. Table 2 shows the calculation for the three measures for the feature "cairo". The examples had been shown using data from tokenization approach A, but the same approach and tables are produced when performing tokenization approach B.

IV. RANKING FEATURES

Given a score for each feature, we are able to go through all of the original tweets and classify the tweet using the feature within the tweet that has the highest score. We keep track of how many times a feature is used. Ordering on times that the feature had been used to predict a class gives us a ranking of all the features. Table 3 shows top features used by the three measures.

Classical Accuracy		Percent Contribution		TF-IDF	
feature	timesUsed	feature	timesUsed	feature	timesUsed
i	15508	i	15233	egypt	284295
square	5461	square	5492	in	87731
im	3547	me	3450	the	54929
me	3487	im	3446	cairo	48124
protesters	2872	protesters	2870	tahrir	39523
mubarak	2603	mubarak	2595	i	37401
revolution	2302	revolution	2352	to	19928
cairo	1603	cairo	1714	rt	14162
tahrir	1410	tahrir	1448	a	12600
egypts	1384	egypts	1395	el	9096
news	1367	news	1375	25-Jan	7062
protests	1359	protests	1346	of	6813
jazeera	1213	jazeera	1213	is	6716
president	1175	president	1180	and	6361
…	…	…	…	…	…

Table 3: Top Features used in the Classification of Tweets (tokenization approach A)

From the table, we see that TF-IDF has identified many stopwords as important because those features are a big percentage of the tweets. We see that percentage contribution used the feature "i" less than classical accuracy (score for feature "i" is slightly less by percent contribution) and the feature "square" more than classical accuracy (score for feature "square" is ranked slightly higher by percent contribution. Percent contribution should be more accurate because it takes into account number of records within each class when calculating its scores.

Table 4: Shows the same type of analysis performed on hashtags and at-mentions (tokenization approach B).

Classical Accuracy		Percent Contribution		TF-IDF	
feature	timesUse	feature	timesUse	feature	timesUsed
#egypt	48353	#jan25	33848	#egypt	165618
#tahrir	30509	#tahrir	32475	#jan25	29356
#jan25	11475	#egypt	27403	#tahrir	11876
#mubarak	11129	#mubarak	10771	#cairo	5682
#cairo	9385	#cairo	10213	@ghonim	3127
#25jan	3996	#25jan	4187	@addthis	2834
@youtube	3668	@youtube	3660	@youtube	2757
@addthis	3002	@addthis	2972	#news	2469
#news	2601	#news	2652	#tcot	1489
@ghonim	2261	@ghonim	2269	#mubarak	1370
#bahrain	1995	#bahrain	1971	#reasons mubarak islate	1341
#freeegypt	1795	#free egypt	1810	@fatmega loman	1317
#yemen	1700	#yemen	1656	@sand monkey	1266
#egipto	1380	#egipto	1374	#bahrain	1207

Table 4: Top Features used in the Classification of Tweets (tokenization approach B)

When looking at top 100 features associated with each class there is a clear difference between classes as we go from class c_0 to class c_9. Features seem to be going from clear topics during the Egypt revolution to features corresponding to personal tweets. Tables 5 and 6 show percent contribution measure top features for classes c_0, c_1, c_8 and c_9 for tokenization approach A and B (respectively).

c0_Feature	c1_Feature	…	c8_Feature	c9_Feature
protesters	square	…	me	i
cairo	jazeera	…	ive	ppl
tahrir	yemen	…	love	anyone
egypts	mubaraks	…	you	so
news	live	…	think	know
president	clashes	…	damn	morning
cairos	http English aljazeera netwatchnow	…	quoti	ok
military	al	…	haha	going
reuters	humidity	…	terradaki	go
25-Jan	algeria	…	whats	swear
hosni	cooper	…	am	omg
thousands	resignation	…	nretweet	khouly3
…	…	…	…	…

Table 5: Top Features Percent Contribution (tokenization Approach A)

c0_Feature	c1_Feature	…	c8_Feature	c9_Feature
#jan25	@youtube	…	@dima_khatib	#fb
#tahrir	#yemen	…	@elazul	@nevinezaki
#cairo	#libya	…	@alyaagad	@etharkamal

#25jan	@ajenglish	…	@mamoudinijad	@gsquare86
@addthis	@waelabbas	…	@khalawa69	@monasosh
#news	#algeria	…	@amrwaked	@nadaauf
#freeegypt	@salmaeldaly	…	@saraaayman	@khouly3
#aljazeera	@ajelive	…	@theonly warman	#icantdateyou
#feb11	#weather	…	#grammys	@travellerw
#alarabiya	@huffington post	…	@terradaki	@mosaa berizing
#ghonim	@shmpongo	…	@litfreak	#iheard
@guardian	@washington post	…	@marionnette90	#mbmemories
@addtoany	@adel_salib	…	@samiyusuf	#prayforegypt

Table 6: Top Features Percent Contribution (tokenization Approach B)

Having found ranked features for each class, the analyst can verify if the features captured make sense and use those features in order to filter and collect more data from Twitter.

V. Analyzing Overall Error

Twitter is a large noisy data source. There are 400 million tweets a day with most of the messages not relevant to the analyst. Simply grabbing a lot of data and trying to fit a model to the messages is not relevant. An analyst needs to first understand how to query Twitter just as an ordinary human being knows how to query the World Wide Web. Querying Twitter is equivalent to understanding the types of features (query terms) to use. We have illustrated a means of ranking features and then using those features for classifying a 10 class problem. Calculating accuracy is simple in the sense that we can just count how many times we have accurately identified a record vs. number of records attempted. The features are ranked by the overall accuracy for the 10 classes achieving 0.695% for percent contribution error, but the final classifier had hundreds of thousands of features that appeared only once. For this reason we choose to look at only the top 1000 features, with accuracies shown in Table 7 and 8.

Classic Accuracy	24.46%
Percent Contribution	24.39%
TF-IDF	15.76%

Table 7: Accuracies for 3 methods using top 1000 features (tokenization Approach A)

Classic Accuracy	23.39%
Percent Contribution	22.70%
TF-IDF	20.67%

Table 8: Accuracies for 3 methods using top 1000 features (tokenization Approach B)

It should be kept in mind that this is a 10 class problem so random guessing would produce around 10% accuracy. TF-IDF actually exhibits worse errors rates because there are few classes so that many features appear in all classes and thus get ranked 0.

The actual accuracy is whether the features that were extracted make sense and can these features be used for finding relevant tweets that are of interest to the analyst. We have seen that the features identified distinguish classes. We saw for instance, that class c_9 carries features that are associated with more personal messages and class c_0 carries features that are closely associated with the Egypt uprising news topics (all other classes are somewhere in between). Going through the messages by hand in the classes we see that c_0 may contain tweets that should not be associated with class c_0, such as:

a) *"OMFGCould you believe it? My wife just purchased an Iphone for 42US$!!! http://moourl.com/5td4g Cairo #famouslies White Stripes DiPietro"*
b) *"WOW OMG JUST WOW -- search Twitter annnd Google side by side - http://bit.ly/hBxUBC #### #ifyouonlyknew Cairo Charles Barkley"*

Likewise other classes probably have tweets that have been misclassified. We use the top 100 features to reclassify the ten classes, but it is up to the analyst to determine if those features are enough (tweets that do not have the features are thrown away).

Among other things that might help in extracting useful tweets and increasing accuracy include analyzing how many times a tweet has been reposted (retweeted), how many people replied to it, and considering the geospatial component so that we focus only on tweets from a certain area. A way to filter irrelevant tweets would be to get rid of tweets that consist of features that are mentioned by less than N number of people (bottom up approach). This can be established through regression to determine a threshold. Another way is to try and understand most important features extracted and include the features from all of the tweets that mention those features (a top down approach). Filtering would get rid of spam and self-centered messages that do not give any insight in understanding the event of interest. In the next section we consider allowing the user to focus on features coming from a specific geo-location.

VI. Visualization

Tweets have a geospatial component to them so that they may be shown on a map. We have used JavaScript and Google Map API in order to visually present Twitter data for the Egypt dataset. The intention was to allow a user to click and drill down into tweets corresponding to some geographic location. In this way a user could perform analysis on how the features in one geographic location are different from tweets associated with a different geographic locations. The thought is that there will be more conversations in the local area where the event is actually taking place then in the rest of the world.

Based on the latitudes and longitudes in the Egypt dataset, we divide the world into a ten by ten grid. This grid serves effectively as a histogram and displays rings that correspond to number of tweets coming from a particular area (the center of the ring is the center for the particular cell on grid and so

some circles may appear on water, Figure 3). The Egyptian government limited Internet access so we actually see that most tweets that do have a geospatial location are from around the world (in particular from South Africa).

For the Egypt dataset, unfortunately only about 1% of data had a geo location associated with it, but this is typical as less than three percent of all tweets have geo-location information [28]. Here the user would select the Egypt province in order to focus on tweets from that area. The top 20 features would be used to filter tweets that don't have a geolocation in order to identify the next top 20 features. This iterative process discovers more and more tweets, avoids spam, and simplifies the computational requirements by not having to consider hundreds of thousands of tweets simultaneously. Results are still to follow in methods to appropriately use geographical information associated with tweets.

The benefit to using geo-location is that the user can focus on main features corresponding to the area of interest vs. discussions about the topic in neighboring regions. For example a victory in a sporting event will be discussed differently in the hometown vs. the rest of the country. The features coming from hometown will probably be positive about the hometown team. These features can then be used for finding towns that have similar feelings about the sport's team. This is an iterative process whereby more and more features can be discovered but at the root of those features will be the features associated with the hometown. The features can be listed in a hierarchical fashion and can be a means of organizing based on features and locations. Unfortunately only 1% of features have geo-locations, again results are still to follow for ranking using this approach.

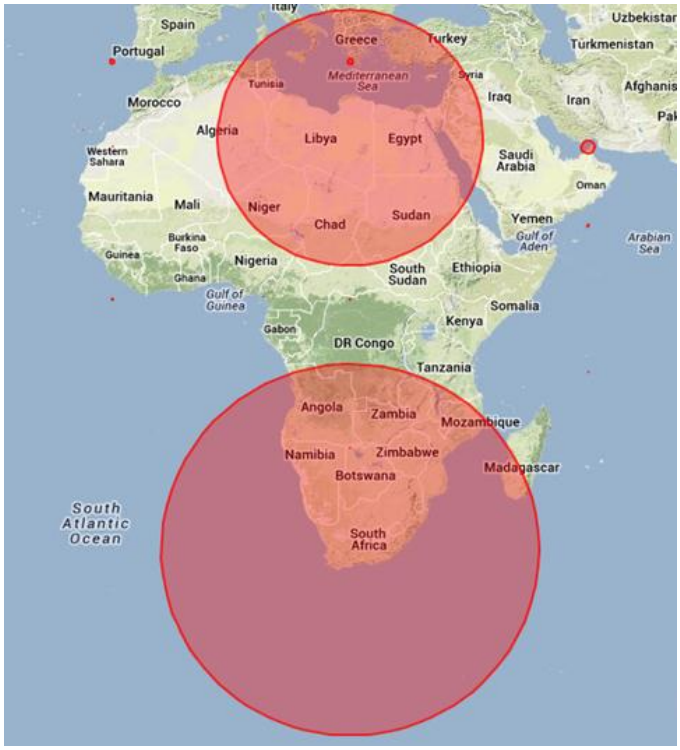

Figure 3. Selecting Tweets based on Geospatial Coordinates

VII. Discussion and Conclusions

In this paper, we explored the use of Twitter as a source of intelligence for determining the status of a pending, emerging, or on-going event. We believe Twitter can be queried for relevant data similar to queries performed on the World Wide Web. In order to make queries, an analyst should have a list of key features (query terms) related to some event of interest. In an attempt to extract those key features, we investigated a 10 pre-labeled class dataset by the ARL covering the Egypt uprising. The features from the labeled classes are used in a frequency matrix so that ranked features can be used to identify other relevant tweets (like Vector Space Model [VSM]). Once top features are identified, an analyst can use an open IE system to make queries for relevant tweets just as a person is searching for web documents on Google. These extra tweets are used to get at an even more robust feature set. Each loop generates a list of features that an analyst has to go through and approve. In this way, we foresee an iterative process between the analyst (feature approver), VSM (feature rankings), and an Open IE system (Twitter queries) in order to create catalogs of useful features for semantic analysis of activities. Catalogs of useful features can then be used for filtering and identifying events and activities of interest.

VIII. Acknoledgements

We thank Sue E. Kase and Liz Bowman at the Army Research Lab and Mike Hinman at the Air Force Research Lab for their help in getting access to the Egypt data.

References

[1] Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

[2] Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18 (11), 613-620.

[3] Weaver, W. (1955). Translation. In Locke, W., & Booth, D. (Eds.), *Machine Translation of Languages*: Fourteen Essays. MIT Press, Cambridge, MA.

[4] Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell System Technical Journal*, 62 (6), 1753-1806.

[5] Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, Vol. 37, Issue 1, pp. 141-188.

[6] Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pp. 315-322.

[7] Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011) Named Entity Recognition in Tweets: An Experimental Study. *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pp. 1524-1643.

[8] Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

[9] Han, B. and Baldwin, T. (2011). Lexical normalization of short text messages: Makn sens a #twitter. In *49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, pp. 368-378.

[10] Gouws, S., Metzler, D., Cai, C., and Hovy, E. (2011). Contextual bearing on linguistic variation in social media. In *Proceedings of the Workshop on Language in Social Media*, pp. 20-29.

[11] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *49th Annual Meeting of the Association for Computational Linguistics*: *Human Language Technologies*, pp. 42-47.

[12] Philips, L. (1990). Hanging on the Metaphone. *Computer Language*, 7(12), pp. 39-44.

[13] Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proc. of Association for Computational Linguistics*., pp. 384-394.

[14] Barbosa, L. and Feng, J. (2010). Robust sentiment detection on Twitter from biased and noisy data. In *Proc. of Conf. on Computational Logistics (COLING)*, pp. 36-44.

[15] Bollen, J., Pepe, A., & Mao, H. (2009). Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. arXiv.org, arXiv:0911.1583v0911 [cs.CY] 0919 Nov 2009.

[16] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pp. 851–860.

[17] Asur, S. and Huberman, B. A. (2010) Predicting the Future with Social Media. *Proc. IEEE/WIC/ACM Int'l Conf on Web Intelligence and Intelligent Agent Technology*, pp. 492-499.

[18] Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam. (2011). Open information extraction: the second generation. In *International Joint Conference on Artificial Intelligence*.

[19] Mausam, Schmitz, M., Soderland, S., Bart, R. and Etzioni, O. (2012) Open language learning for information extraction. In *Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523-534.

[20] Gamallo, P. Garcia, M., and Fernandez-Lanza, S. (2012) Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pp. 10-18.

[21] Beaumont, P. (2011) The truth about Twitter, Facebook, and the uprisings in the Arab world. *The Guardian* (Feb. 25, 2011); http://www.guardian.co.uk/world/2011/feb/25/twitter-facebook-uprisings-arab-libya

[12] Ghonim, W. (2011) Interviewed by Harry Smith. Wael Ghonim and Egypt's new age revolution. *60 Minutes* (Feb. 13, 2011); http://www.cbsnews.com/stories/2011/02/13/60minutes/main20031701.shtml?tag=contentMain;contentBody

[22] Al Jazeera. Timeline: Egypt's revolution (Feb. 14, 2011); http://english.aljazeera.net/news/middleeast/2011/01/201112515334871490.html

[24] Kase, S. E. and Bowman, L. ARL Egypt Twitter Data Set, 2012.

[25] Hongzhao Huang, Arkaitz Zubiaga, Heng Ji, Hongbo Deng, Dong Wang, Hieu Khac Le, Tarek Abdelzaher, Jiawei Han, Alice Leung, John Hancock and Clare Voss. 2012. Tweet Ranking based on Heterogeneous Networks. Proc. 24th International Conference on Computational Linguistics (COLING2012).

[26] Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In Proceedings of EMNLP, volume 4. Barcelona: ACL.

[27] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28 (1), 11-21.

[28] Leetaru, K., Wang, S., Cao, G., Padmanabhan, A., and Shook, E. (2013). Mapping the global twitter heartbeat: The geography of twitter. *First Monday*, 18(5).