Microsoft Research Social Media Collective Report
MSR-SMC-11-01

LEARNING TO WORK WITH LARGE-SCALE TWITTER DATA SETS: USING OFF-THE-SHELF TOOLS TO QUICKLY AND EASILY SEE TWEET PATTERNS

A. Marwick[1] , J. Gonzales-Rivero [1,2]

1 *Microsoft Research New England*
1 Memorial Drive, 12th Floor, Cambridge MA 02141
amarwick@microsoft.com

2 *Franklin W. Olin College of Engineering*
Needham, MA 02492
Jazmin.Gonzalez-Rivero@olin.edu

October 2011

1. Introduction

With the increasing popularity of large social software applications like Facebook and Twitter, social scientists and computer scientists have begun developing innovative approaches to dealing with the vast amounts of data produced and collected in such environments. For qualitative researchers, the methods involved can be daunting and unfamiliar. For quantitative researchers, using "big data" to answer complex social queries can be challenging. In this report, we outline some basic procedures for working with a large-scale Twitter data set to answer qualitative inquiries. This paper is aimed at social scientists and humanities scholars with limited experience with big data and a lack of computing resources to do extensive quantitative research.

2. Social Media, Social Science

The data that we used was collected as part of a larger project conducting ethnographic audience research on how young people understand gay and lesbian relationships on the FOX Network musical series Glee.[1] Glee has been lauded by critics and activists for its ground-breaking portrayals of young gay, lesbian, and bisexual teens. For its 2010-2011 season, the show was nominated for 4 Emmys and 5 Golden Globes and won the 2011 Gay & Lesbian Alliance Against Defamation (GLAAD) Award for Best Comedy.

[1] Besides Alice Marwick, the principle researchers on this project are Mike Ananny (Microsoft Research) and Mary Gray (Indiana University, Bloomington).

Glee is generally considered a “transmedia” text (Jenkins, 2006) in that the show’s story, while primarily told through the television series, is supplemented with albums and mp3s of cast recordings, a concert tour, and a 3D movie of the concert tour, in addition to various Facebook, Twitter, and YouTube channels maintained by official sources, cast members, and fans. To understand how young people think about and use the show, the project researchers wanted to conduct a multi-sited audience study that took these different interactions into account (Marcus, 1995). To that end, the project involved ethnographic observation of a group of college students watching Glee together over a five-week period, as well as individual interviews with the student watchers. To supplement this qualitative data, the researchers turned to social.

Twitter is a major site used for fan discussion and celebrity interaction (Marwick & boyd, 2011). On Twitter, a hashtag is an informal content ordering scheme used to find related content by appending the pound sign to a topic, such as #chi2011 or #stateoftheunion (Romero, Meeder, & Kleinberg, 2011). We found three prominent hashtags used in Glee-related discussions. The first one, #glee, is used for general discussion of the show. The second and third, #klaine and #brittana, are portmanteaus of two couples, “Kurt and Blaine” and “Brittany and Santana,” respectively. Kurt and Blaine are teenage boys, while Brittany and Santana are teenage girls. By examining each, we hoped to identify any variance between how “gay” and “lesbian” relationships are taken up and used by viewers.

We collected all tweets including any of these three hashtags over a nine week period, which resulted in a corpus of approximately 450,000 tweets. The script that was used to collect the #glee data accessed the Twitter Search API every five minutes. The script queries for tweets that include one of the wanted keywords (#glee, #brittana, #klaine), makes sure it doesn't store the same tweets multiple times, and saves them in a MySQL database.

At this point, we needed to come up with creative ways of parsing this enormous data set using easily accessible tools. Gonzales-Rivero was a summer intern with our group. She, Alice Marwick, and Mike Ananny devised some simple methods using off-the-shelf tools.

3. Tool Selection

We used MySQL to work with the tweets. MySQL was used because it is easily compatible with Python, PHP, and Excel. Given that Gonzales-Rivero was only with the team for 8 weeks, we wanted a tool without a steep learning curve that could be quickly learned by someone who had never used it before. Since MySQL is widely used, there is plenty of documentation and a helpful and active community.

Gonzales-Rivero decided to use Python because it is easily compatible with MySQL and an easy and quick program to install. One can use various scripting and non-scripting languages to parse text documents.

4. Working with Tweets and MySQL

The first thing we needed to do after opening the database in MySQL was to break it down into smaller chunks for analysis as there were few ways we could productively analyze 450,000 tweets. We needed to divide the entire corpus into subgroups, both to make the data more manageable to work with and to answer questions we had about differences between perceptions of the two couples on the show.

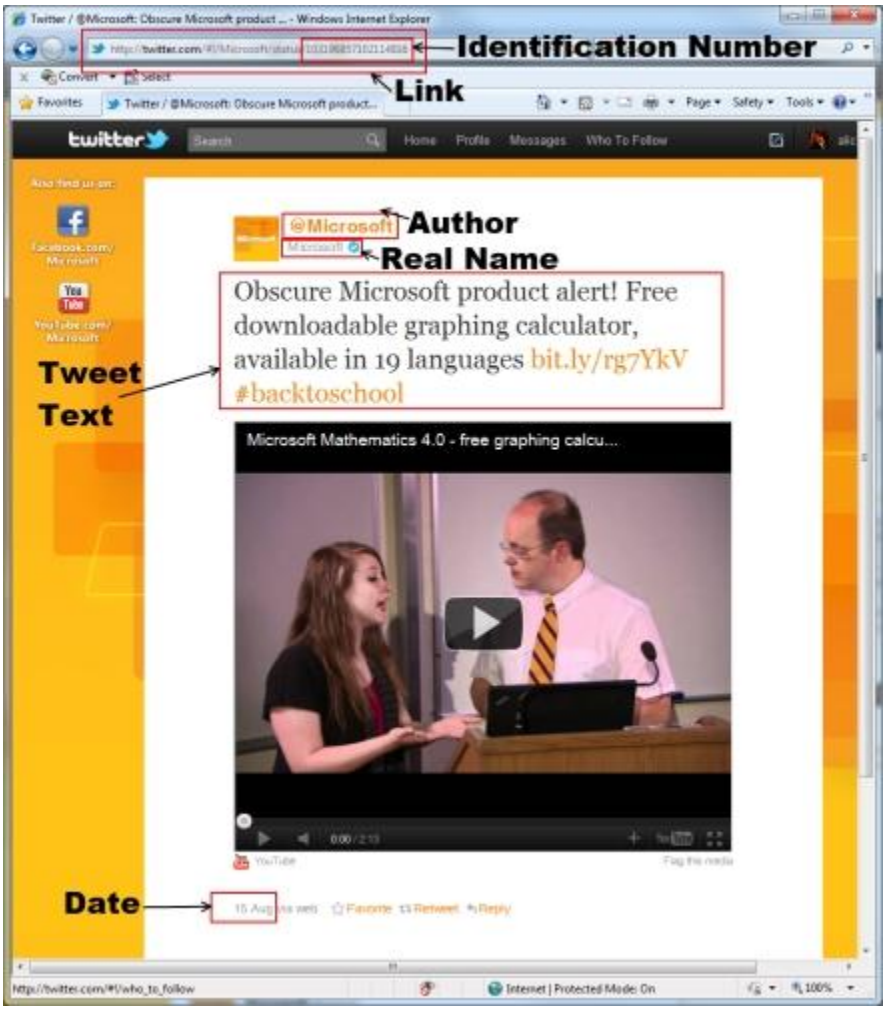

Figure 1: Tweet Parts

Each tweet we collected had several parts: ID number, tweet text, link, author, realname, and date time. You can see where each of these appears on an actual tweet page in Figure 1.

ID Number	Unique identifier for each tweet
Tweet text	The content of the tweet
Link	The tweet's URL
Author	Twitter handle of the author
Realname	The author's "real name," displayed below their Twitter handle on individual tweets and profile pages
Date time	The time the tweet was sent (including day, hour, and second).

For different queries, we worked with different parts of tweets.

Other fields were generated by the tool used to collect the Twitter data, but we did not use these in our analysis.

4.1.Creating Subgroups

Gonzales-Rivero started with two subgroups: the tweets containing the hashtag "#klaine" and the tweets containing the hashtag "#brittana". She created a table of each subgroup by searching the "tweet text" portion of each tweet for #klaine or #brittana.

```
[mysql> CREATE TABLE klaine LIKE alicepaper;
mysql> INSERT INTO klaine SELECT * FROM alicepaper WHERE text
LIKE "%#klaine% OR text LIKE "#klaine%" OR text LIKE "%#klaine"
OR text LIKE "#klaine"; ?]
```

The query copies the structure of the main table (id number, tweet text, pubdate, link, author, realname, storyquery, date time) and then queries to the main database: copy all the information that has the hashtag "#klaine" into the query.

This resulted in three tables: the first table containing all the tweets, one table containing all the tweets with #klaine, and another with all the #brittana tweets. At this point, we could delve deeper into the data.

4.2.Working with Timelines

We were curious when people tweeted about Glee. Did they tweet while the episode was airing, when it was released for free streaming the next day on Hulu.com, or at times that didn't correspond with episode air dates? Did tweets about different couples correspond with particular episodes?

When trying to create a timeline, the only part of each tweet that was relevant was the date. We created a CSV file that contained solely the date and the number of tweets sent on that date.

```
[mysql> SELECT pubdate, COUNT(pubdate) FROM alicepaper GROUP BY
pubdate ORDER BY pubdate INTO OUTFILE "timeline.csv" FIELDS
TERMINATED BY ",";]
```

Twitter defines date with an hour, second, and minute value [2011-04-13 14:30:38]. If you keep the date in this format, however, the query will only find tweets that were sent on the exact same minute. Instead, we wrote the query to look specifically at the "day" or the "hour" in order to get a manageable, but granular level.

Gonzales-Rivero began by making a few empty tables. For each category (overall data, retweet[2], klaine data, and brittana data) she made three tables. One table would eventually hold the exact date down to the second and the number of tweets corresponding to that date. Another held the date up to the day, and the number of tweets corresponding to that day. The final table held the date down to the hour and the number of tweets corresponding to that hour. After making these tables, Gonzales-Rivero wrote the following procedure.

```
[Mysql> DELIMITER $$
Mysql> CREATE PROCEDURE OverAllHour()
-> BEGIN
DECLARE d INT Default 0;
DECLARE h INT Default 0;
DECLARE m INT;
SET m = 4;
Simple_loop: LOOP
IF h != 25 THEN
SET h = h + 1;
INSERT INTO overallhourly SELECT month, day, year, hour,
COUNT(num_tweets) FROM overalldates WHERE month = m AND day = d
AND hour = h;
ELSEIF d != 31 THEN
SET h = 0;
SET d = d + 1;
ELSEIF m != 6 THEN
SET m = m + 1;
SET h  = 0;
SET d  = 0;
ELSE
LEAVE simple_loop;
END IF;
END LOOP simple_loop;
END $$
DELIMITER ;
```

[2] See section 4.3 for more on working with retweets.

```
CALL OverAllHour();
SELECT * FROM overallhourly INTO OUTFILE "hourlytimeline.csv"
FIELDS TERMINATED BY ",";
]
```

The data had to be added to a table previous to exporting: if the exporting line had been added into the loop itself, the program would try to create a new document with the same name every time it went through the loop. Note that the table "overalldates" contains the full dates and the number of tweets corresponding to each date, and that the table "overallhourly" will contain the date up to the hour, and the number of tweets corresponding to that hour. Code for each category looked essentially the same except for the names of the tables.

We then used Excel to create a graph of when tweets were taking place (Figure 2).

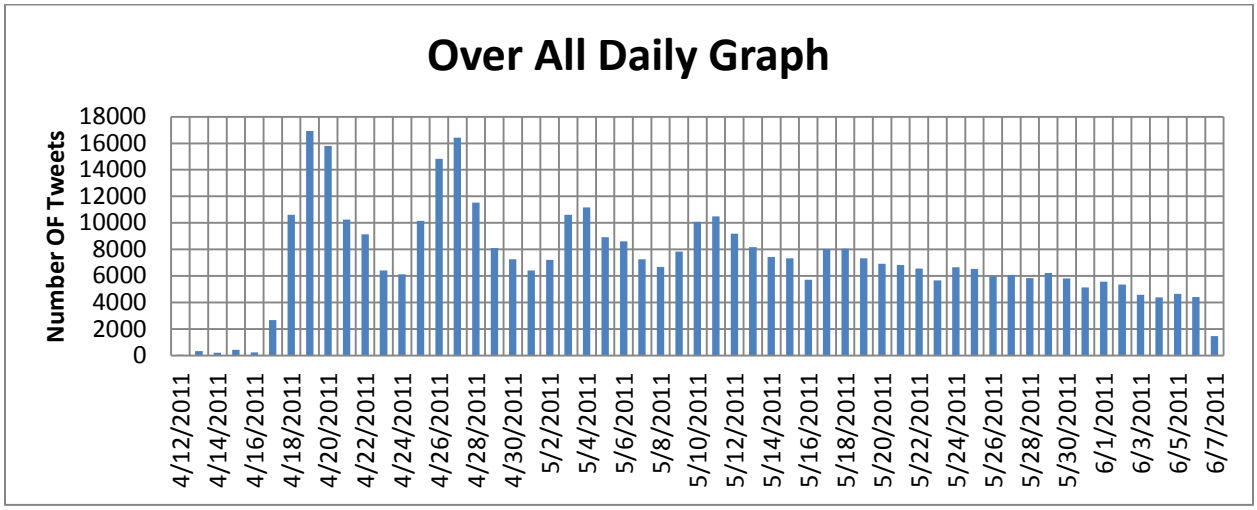

Figure 2: Tweets over Time

4.3.Working with Users and Retweets

We were interested in seeing which users posted the most, and which were re-tweeted the most. Re-tweeting is "the Twitter-equivalent of email forwarding where users post messages originally posted by others." (boyd, Golder, & Lotan, 2010). Retweets (RT) can be identified because they typically contain the capital letters "RT." For example:

```
@pj_williamson Yesssss!!! RT @JennaJatUK: How is it that #Glee
manages to make so many songs better than the originals?
#rockingoutintheoffice
```

In this tweet, user pj_williamson re-tweets a tweet originally posted by user JennaJatUK. He adds "Yessss!!!" to show his agreement. Users frequently alter RTs to add commentary, to

shorten them to fit within Twitter's 140 character limit, or for a variety of other reasons. This makes it difficult to determine, for example, which tweets were most re-tweeted.

To determine who was *retweeting* the most, we created a table of RTs, using the same method we made the #brittana and #klaine tables. Because the letters "rt" appear in many common words, we searched for "RT @," since RT was always followed by a Twitter user's handle. We queried MySQL to return the author and how many times they posted.

To figure out *who* was retweeted the most, we used Python to parse the tweet text to examine individual words in the tweet texts, and return the most frequently used. When a user is retweeted, their username is preceeded by "RT @". The code removed all words not preceded by "RT @" this returned a file of @symbols with a count of each.

To determine which users were tweeting the most, we simply queried MySQL to return the author and how many times they posted.

5. Word Counts and Wordles

One of our main research questions involved understanding how people perceived the gay and lesbian couples on the show. To determine the overall valence around each couple, we first determined the most frequent words used in each #hashtag.

We used Python to create a large text file containing only the tweet text fields (without date, author, or ID number). Gonzales-Rivero wrote a query in Python to parse the file one word at a time. As Python found new words, they were added to an array. If the word had already been found, the count was raised. This returned the top words, but also included words like "like" and "I" which appear frequently. In order to work with the @symbol, we could not remove the punctuation.

Next, we used Wordle (http://www.wordle.net/), a project from IBM Research that creates "word clouds" out of a corpus. Word clouds are a great tool to use to visualize a large amount of data in an easy-to-read format, and make quick and obvious discoveries.

We entered the text files generated by Python for each hashtag and generated a word cloud (for the very large files, we used the top 700 words). For example, this is the word cloud for the first week of tweets:

Figure 3: Common Words in a Sample of Glee Tweets

You can see that "brittana," one of the hashtags, is very prominent in this sample, as is "RT." Character and actor names, like "charice," "Santana," "kurt," and "Morrison" are interspersed with references to other Twitter accounts (@gleeks, @ultimategleeks, @officialcharice). We were surprised to see that the #brittana hashtag was so prominent compared to the #klaine hashtag as we expected that the Kurt-Blaine pairing would be more popular; they appear in more episodes, appear in the press more often, and were regularly mentioned in our interviews.

We made three types of Wordles:

- "Regular" which included all words except for the main word (in most cases, #glee).
- "RT" which contained only retweets.
- "Weekly" Wordles, which contained only the data from that pertaining week.

Wordle removes most punctuation, including semi colons, commas, periods, and question marks, so emoticons are not included.[3]

The Wordle program automatically removes commonly used words in one specific language. In our case, it removed common English words like "the," but not the equivalent words in other languages that appeared in our corpus that used the Latin character set, specifically Spanish or

[3] Luckily, Wordle does not strip out @ signs.

Portuguese. As a result, common Spanish words like “de” appear overly prominently in the Wordles.

6. Helpful Hints

Gonzales-Rivero has a few tips for people relatively new to programming who are working with similar types of data.

- The titles of every document you generate must be very clear (csv files, txt file, tables, Wordles, variables and procedures). It’s easy to forget what you’re doing when you’re working with a large corpus for several weeks.

- Choose “Latin1” as the language set in Python and MySQL. The default character set will be confused when you run across characters like tildes (which appear frequently in Spanish and Portuguese tweets), creating a string of randomness.

- The Twitter database holds more data than you’d think, making tools that can be used to look at the database as a whole very valuable. It is impossible to print out the entire database and look through it manually. Even if it means writing a few extra lines of code, this is far more efficient then doing things by hand or even partially by hand.

- The date/time in Excel is not the same as the date/time in MySQL, and it must be translated. When exporting from MySQL the date/time is automatically translated, but when going from Excel to MySQL we ran the following code where A1 contains the excel version of the date.

```
=CONCATENATE(CONCATENATE("",YEAR(A1)),IF(LEN(MONTH(A1)) =
1,CONCATENATE("0",MONTH(A1))),IF(LEN(DAY(A1)) =
1,CONCATENATE("0",DAY(A1)),CONCATENATE("",DAY(A1))))
```

- When referring to a document in MySQL, you must add a double backslash between folders for it to work correctly. For example “C:\\Users\\ jagonz\\Glee Project\\Jazmin Data\\Graph Data\\OverAllText.txt”

7. Conclusion

The data generated by contemporary social media applications contains a great deal of valuable information, but it can be challenging to extract this information. We hope this document is useful to other researchers embarking on similar endeavors. More about the work of our team at Microsoft Research, the Social Media Collective, can be found on our blog, http://www.socialmediacollective.org.

8. References

boyd, danah, Golder, S., & Lotan, G. (2010). Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter. *Proceedings of the Forty-Third Hawai'i International Conference on System Sciences (HICSS-43)* (pp. 1–10). Presented at the HICSS-43, Kauai, HI: IEEE Computer Society.

Jenkins, H. (2006). *Convergence culture*. New York: New York University Press.

Marcus, G. E. (1995). Ethnography in/of the world system: the emergence of multi-sited ethnography. *Annual Review of Anthropology*, *24*, 95–117.

Marwick, A., & boyd, danah. (2011). To see and be seen: Celebrity practice on Twitter. *Convergence*, *17*(2), 139-158.

Romero, D. M., Meeder, B., & Kleinberg, J. (2011). Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. *Proceedings of the 20th international conference on World wide web* (pp. 695–704).