

Mining Twitter Using R

Presented to the St. Louis R Users Group

By Mathew Woodyard

April 3, 2012

1 Motivation

We endeavor to answer the questions: Why aren't people buying my widgets? Are people saying bad things about me? Why are the students of Southern Illinois University Edwardsville (SIUE) unhappy?

These are classic questions concerning not only product marketers, but statisticians. What convinces people to tell us what they think, what they own, what they will buy, where they are, and who they are? How do we get honest answers? What can we infer from a large number of these answers?

There is some good news yet because...

1.1 300 Million People Are Here To Chat!

Twitter is a “microblogging” service that facilitates communication. Each message, a “tweet”, contains no more than 140 characters. 300 million people use Twitter. But how do they use it? And why?

Twitter has some use cases and features that result in a nice corpus. Twitter is often used for consumption of news stories and participation in current events. Many people also use twitter to express emotional states: often complaining about and praising places, products, and services. Since tweets contain no more than 140 characters, finding relevant tweets is normally a fairly easy task. Twitter's automatic grouping and “hashtags”, tags that group topics and start with a hash mark (#), enable us find what we want with relative ease.

In my example, I will use tweets about SIUE. Tweets were selected from the stream based on the occurrence of the terms “SIUE” and “#onlyatsiue”, the latter having spent some time as a major regional trending topic.



Figure 1. A tweet in its natural habitat.

Now that we know people are generating data we can use, how can we provide analysis that makes these tweets useful?

2 Harnessing Tweets

Our strategy will consist of the following steps.

1. Sip from the stream of tweets using `twitterR` and `ROAuth` to build a corpus.
2. Discover your new corpus.
3. Select a sentiment lexicon. Download it and load it into R.
4. Score tweets using the function provided or using your own classification scheme.
5. Summarize the results.

2.1 Building a Corpus

First, we sip from the stream to build a corpus. Be aware that we can only sip, and not guzzle, due to Twitter's rate limiting (note there is no limit on the number of tweets you post using `twitterR`). The code below loads some previously-downloaded tweets. Code to append new tweets from the stream is commented out and can be ran when you want to expand the corpus. Note that by registering with Twitter's developer program, you can get API keys used to increase the rate limit for your IP/key combination.

```
> # Load needed libraries to allow R to interface with twitter and
> # interface with the API using OAuth.
> require(twitterR)
> require(ROAuth)
> # The code below does not work due to a current bug in twitterR.
> # This means that the rate of downloading tweets will be limited.
> # The following code should NEVER be shared. It includes private keys
> # that authenticate against the twitter API. This code was taken from the
> # twitterR documentation. Run the code from this block to the next comment.
> #
> # reqURL <- "https://api.twitter.com/oauth/request_token"
> # accessURL <- "https://api.twitter.com/oauth/access_token"
> # authURL <- "https://api.twitter.com/oauth/authorize"
> # consumerKey <- "keygoeshere"
> # consumerSecret <- "keygoeshere"
> # twitCred <- OAuthFactory$new(consumerKey=consumer.key,
> #                               consumerSecret=consumer.secret,
> #                               requestURL=reqURL,
> #                               accessURL=accessURL,
```

```

> #                                     authURL=authURL)
> # twitCred$handshake()
> #
> # After entering the PIN provided by twitter, run the following command.
> #
> # registerTwitterOAuth(twitCred)
>
> # Load saved tweets.
> load(file="../..//siue.tweets.RData")
>
> # Uncomment to add hot new tweets.
> # siueTweets <- append(siue.tweets, searchTwitter('siue', n=1500))
> # siueTweets <- append(siue.tweets, searchTwitter('#onlyatsiue', n=1500))

```

2.2 Exploring Your Hot New Corpus

Now that we have the corpus loaded, let's explore what people from SIUE are saying about their university.

```

> tail(siue.tweets, 2)

[[1]]
[1] "ktdare: Fire alarm? Check. Short power outage? Check. #onlyatsiue"

[[2]]
[1] "ChrisHeinle2211: thought a goose was going to attack me #OnlyAtSiue"

```

So far we have some anecdotes and can probably infer that their university had a power outage, is a goose sanctuary, and has a bit of a wasp problem. But what kind of sentiments are associated with SIUE in the aggregate?

2.3 Loading Lexicons; Preparing Tweets

Before we can perform more analysis on the aggregate sentiments within these tweets, we need to first download and load a sentiment lexicon into R. Then we need to modify the corpus so that we can analyze it.

The Sentiment Lexicon for this project is taken from Hu and Liu's opinion lexicon. See <http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html> for source information and <http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar> to download their opinion lexicon. The raw text is also included in the Git repository for this project.

```

> # Import sentiment lexicons. If the situation demands,
> # add in domain-specific words.
> positiveWords <-
+   scan("../..//Hu and Liu Sentiment Lexicon/positive-words.txt",

```

```
+      what='character', comment.char=';')
> negativeWords <-
+   scan('.../..Hu and Liu Sentiment Lexicon/negative-words.txt',
+       what='character', comment.char=';')
```

Now we need to restructure the corpus so we can preform the analysis as preformed in this project.

```
> # Extract the text of the tweet for mining.
> tweetText <-lapply($tweets, function(x) x$getText())
> # Remove non-UTF8 characters.
> tweetText <- subset(tweetText, !grepl("[\x80-\xFF]", tweetText))
```

2.4 Scoring Tweets

The function we are using is mainly taken from “R by example: mining Twitter for consumer attitudes towards airlines” by Jeffrey Breen. If you want to see the function, simply look at the R code in the GitHub project or in the Rnw file that comes with this PDF.

```
> tweetScores <- score.sentiment(tweetText, positiveWords, negativeWords)
```

2.5 Summarizing Your Results

Now we summarize the results of our mining. This is a pretty simple way to summarize the scores, but it gives the insight that tweets about SIUE from this sample were very slightly positive.

```
> # Summarize and graph histogram with ggplot2.
> summary(tweetScores$score)
```

```
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.0000  0.0000  0.0000  0.1426  0.0000  5.0000
```

3 Limitations

While we have done lots of stuff, R isn't perfect and this approach is simplistic.

3.1 Bags of Words Approach

- Irony and function used. Give example.

3.2 Problems

Problems go here. Like limitations of R and stuff.

```
> require(ggplot2)
> qplot(score, data=tweetScores, geom="histogram", binwidth=1)
```

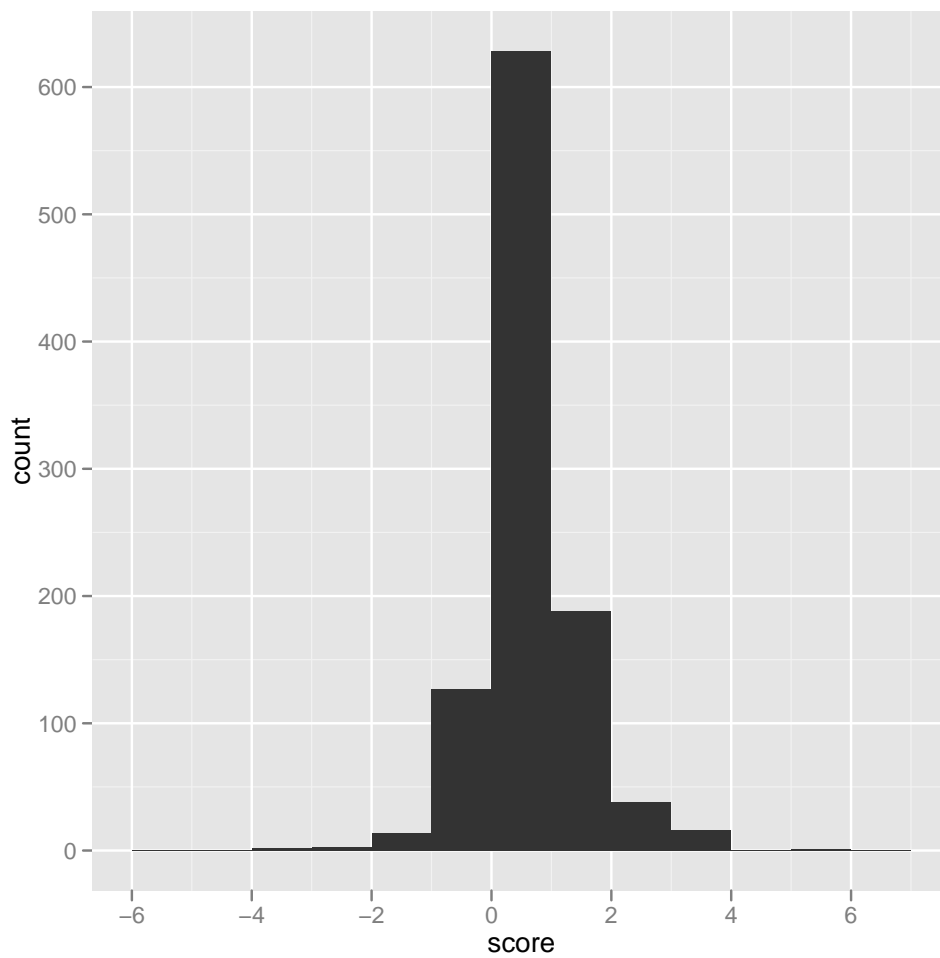


Figure 2. Histogram of tweet scores.

3.3 Requirements

For this project, we stand on the shoulders of giants (as always the case when using R). We have received aid and comfort from the packages `plyr`, `stringr`, `twitterR`, `ggplot2` and `ROAuth`. You will need to have these installed for the code to run. For more information on these packages, see their associated help pages and vignettes.

4 License

“Mining Twitter Using R” by Mathew Woodyard is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States License. My GitHub project found at <https://github.com/woodrad/Twitter-Sentiment-Mining> is under the same license, with the exception of the documentation written by others, which is under the (Artistic) license listed in the documentation. Permissions beyond the scope of this license may be available at <http://5lbs.org>.