Analyzing tweets to detect social media events

Frederick J. Boehm and Bret M. Hanlon

June 15, 2020

Introduction

Twitter has profoundly changed how we communicate. In only 280 characters, users can instantly contribute to public conversations on politics, current events, sports, media, and many other topics. Recent development of accessible statistical methods for large-scale text analysis now enable instructors to use tweets as contemporary pedagogical tools in guiding undergraduate research projects. We report one instance of a mentored text analysis research project. We share our data and computer code to encourage others to undertake tweet text analysis research. We also describe our methods for creating a collection of tweets.

Some social media data, including tweets from Twitter, is available through website application product interfaces (APIs). By way of a streaming API, Twitter shares a sample of approximately one percent of all tweets during an API query time period ("Sampled Stream," n.d.). Any Twitter user can freely access this one percent sample, whereas access to a larger selection is available to researchers for a fee.

Using large collections of tweets, scholars have studied diverse research questions, including the inference of relationships and social networks among Twitter users (Lin et al. 2011); authorship of specific tweets when multiple persons share a single account (Robinson 2016); and rhetoric in recruiting political supporters (Pelled et al. 2018; Wells et al. 2016). Recognizing the potential utility of tweets for data science research and teaching, we created a collection of tweets over time by repeated querying of the Twitter streaming API.

In line with recent calls for students to work with real data (Nolan and Temple Lang 2010), our collection of tweets has served as a valuable resource in our mentoring of undergraduate data science research. Working with real data allows students to develop proficiency not only in statistical analysis, but also in related data science skills, including data transfer from online sources, data storage, using data from multiple file formats, and communicating findings and their limitations. Collaboratively asking and addressing novel questions with our collection of tweets gave mentored students opportunities to develop competency in all of these areas.

Mentoring in the work place and in higher education has many benefits, including improving 1) students' development as thinkers and scholars, 2) confidence in their own abilities, 3) integration into the campus community, and 4) interest in graduate training (Baker and Griffin 2010; Higgins and Kram 2001).

While our tweet collection enables us to address many possible research questions, the dynamic content of tweets over time particularly piqued our interest. Together, students and mentors hypothesized that high-profile social media events would generate a high volume of tweets, and that we'd detect social media events through changes in tweet topic content over time. We present below 1) an approach for collecting tweets in real time and 2) statistical methods for detecting social media events via latent Dirichlet allocation modeling of collections of tweets and 3) reflections on using this data set in research mentoring of undergraduate students.

Methods

Collecting tweets over time

We include here instructions for creating a tweet collection. First, we created a new account on Twitter. With these user credentials, we used the R package `rtweet` to query the API. Because we work with linux operating systems, we use the `crontab` software to repeatedly execute R code to submit API queries. Each query lasts a user-specified duration. We time the API queries so that there is no time lag between queries. We store API query results in their native JSON format. The R package `rtweet` provides functions that parse tweet JSON to R data frames. We then conducted all further analyses in R.

Querying Twitter API to get complete tweets

We queried a tweets database, created with the methods described above, to get ID numbers for tweets from the desired time periods. We then submitted API queries to Twitter to get the full content of the tweets, including the tweet text. We provide below the R code that we used to query the Twitter API to obtain full tweet content.

```r
rtweet::lookup_tweets()
```

Tweet structure

Tweets are available as Javascript Object Notation (JSON) objects. Every tweet consists of multiple named fields, each of which is a key-value pair. The number of fields per tweet depends on user settings, retweet status, and other factors.

PLACE TWEET JSON HERE ### Parsing text of tweets

We used functions from the `rtweet` package to parse tweet JSON into a data frame. From there, we used `tidytext` R package functions to break the tweet text into individual words. We discarded commonly used "stop words" and emojis.

Latent Dirichlet allocation models require that the corpus be inputted as a document by term matrix. Each row corresponds to a single document (a single tweet), and each column is a single term (or word). Each cell contains a count (the number of occurrences of a term in the specified document). We created a document by term matrix with the R functions `from the` R package.

Latent Dirichlet allocation

Latent Dirichlet allocation is a statistical method for inferring latent (unobservable) topics (or themes) from a collection (or corpus) of documents. It is a probabilistic, generative model for a corpus of documents. It is generative in the following sense. We pretend that there's an imaginary process for creating documents in the corpus. For each document, we choose a discrete distribution over topics. For example, some Mother's Day tweets wish mothers a happy celebration. This may constitute one topic in the corpus. Having chosen a distribution over topics, we then select document words by first drawing a topic from the distribution over topics, then drawing a word from the chosen topic. Thus, a topic is technically defined as a distribution over words in a fixed vocabulary (or collection of words).

The literature on latent Dirichlet allocation and related methods is vast, and we won't attempt to review it here.

Study design

We sought to validate our hypothesis that we could detect a major social media event by examining tweet topic content at distinct time periods. As a proof of principle of our event detection strategy, we chose to analyze tweets during and after Mother's Day 2020. We fitted latent Dirichlet allocation models for each of four distinct five-minute periods. The first period began at noon Eastern time on Mother's Day 2020. Subsequent time periods started 24, 48, and 72 hours later. We defined each time period to be a single collection, or corpus, of tweets. We then fitted latent Dirichlet allocation models to each corpus.

We used several criteria to evaluate latent Dirichlet allocation model fits, with emphasis on choosing a reasonable number of topics per corpus. Our strategy involved both visualization and more quantitative approaches to model evaluation. For every model, we created one word cloud per topic.

We then inspected topic contents at each of the four time points.

Results

We applied the project framework to our mentoring of two students. Both engaged in 12 months of research during their senior year of undergraduate studies in statistics. Below, we describe three categories of outcomes:

1. student outcomes
2. mentor outcomes
3. scholarly outcomes

Student Outcomes

We subjectively assessed student outcomes through conversations in our weekly student research meetings. Both students showed increases in confidence and ability to do data science research.

Both students secured positions in data science after graduation. One student enrolled in a statistics graduate program, while the other pursued employment in health care analytics.

Students benefited from our emphasis on the four central concepts, three from Nolan and Temple Lang (2010) plus reproducible research skills. The research projects successfully drew on emerging areas of statistical computing, namely text analysis. They combined computational topics, including topic modeling and time series methods, with data analysis in the practice of statistics. Although we didn't formally measure them, our informal assessment indicates that students' computational reasoning skills increased over the duration of our projects. Students used a variety of computing tools and methods to arrive at a practical solution to a selected task. They became more skilled in computing with R and shell scripts and more fluid in their verbal explanations during our regular meetings (R Core Team 2019).

Our framework's emphasis on reproducible research skills is evidenced by the students' R package, `parseTweetFiles`, which is both version controlled with `git` and shared via Github.com.

Mentor outcomes

We grew as mentors during our work with the two students. We successfully guided junior scientists through a productive, hands-on research experience, and we anticipate refining the framework in future iterations.

Scholarly outcomes

Our scholarly contributions include the `parseTweetFiles` R package on Github (https://github.com/rturn/parseTweetFiles) and presentations at conferences such as useR! 2016 (R users' conference) and local poster sessions. Additionally, both students prepared end-of-project reports on their research.

Discussion

Benefits of our framework

The student test cases for our framework demonstrated greater self-confidence and greater proficiency in data science skills over the course of the research projects. They used real-world data sources to address real scientific research questions. Additionally, they showed great interest in quantitative and data science careers. After graduation, one student immediately enrolled in statistics graduate training, while the other sought employment in health care analytics.

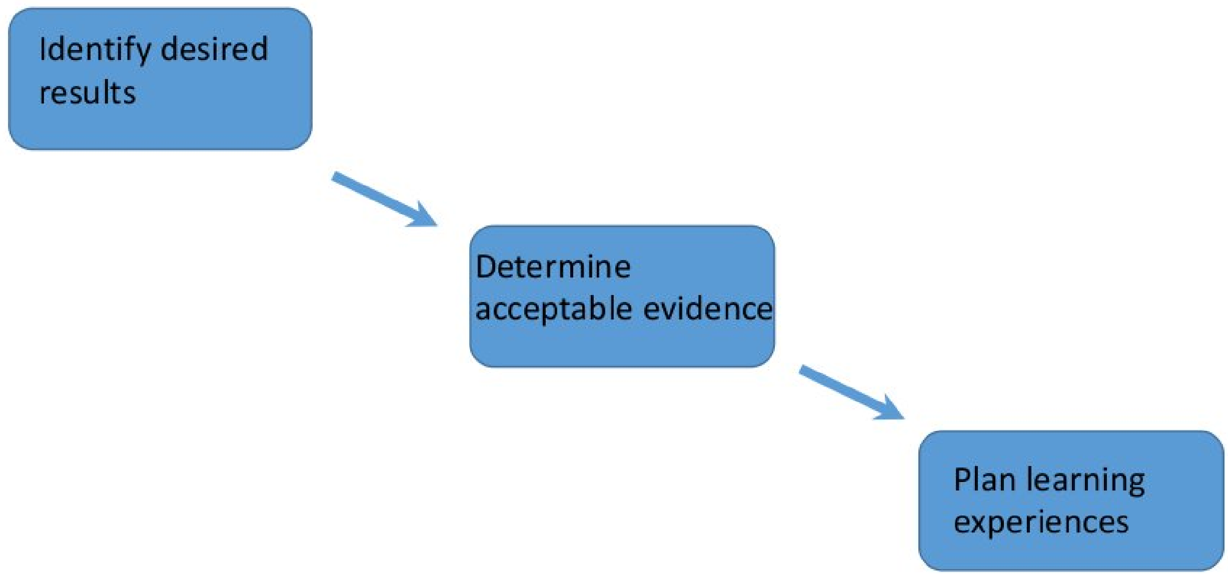
Critiques of our framework

From our current perspective, we offer a number of framework critiques and opportunities for improvement. Our measure of students' self-confidence in research ability was merely subjective. In future iterations of our framework, we would like to measure systematic and objective outcomes. One strategy for implementing this is to administer a survey, including questions from Vance et al. (2017), both before and after the mentored research project. We would use survey questions that focused on student beliefs about themselves, their skills, and their future careers.

One shortcoming of our initial framework was the relative lack of emphasis on best practices for computational reproducibility. This is one area that we would like to rectify in future mentoring activities. The university has periodically offered a semester course in best practices for computationally reproducible research. We especially see collaborative version control systems, such as Git and Github, as essential tools for the modern data scientist.

Framework development with backward design

In future research, we will continue to develop our framework for undergraduate data science research by explicitly incorporating backward design principles (Wiggins and McTighe 2005). Following Wiggins and McTighe (2005), we will identify desired results, determined acceptable evidence, and planned learning experiences.



Before identifying desired results, we will prioritize topics from Nolan and Temple Lang (2010). Specifically, we will assign all terms from Figure 1 of Nolan and Temple Lang (2010) into one of three categories:

1. worth being familiar with
2. important to know and do
3. enduring understanding

We've tabulated below the Nolan and Temple Lang (2010) terms for the current framework and its student projects.

Prioritizing Key Terms from Figure 1 of @nolan2010computing

xxx

Term	Circle
R packages	Enduring understanding
debugging	Enduring understanding
shell tools	Enduring understanding
reproducible computation	Enduring understanding
text editors	Enduring understanding
version control	Enduring understanding
file system concepts	Enduring understanding
text processing	Enduring understanding
regular expressions	Enduring understanding
EM	Important to know and do

MCMC	Important to know and do
Bayesian computation	Important to know and do
programming scope	Important to know and do
data structures	Important to know and do
portability	Important to know and do
authoring tools	Important to know and do
GUIs	Important to know and do
grammar of graphics	Important to know and do
composition	Important to know and do
linear algebra decompositions	Worth being familiar with
representation of numbers	Worth being familiar with
RNG	Worth being familiar with
optimization	Worth being familiar with
numerical algorithms	Worth being familiar with
efficiency	Worth being familiar with
parallel computing	Worth being familiar with
modeling language	Worth being familiar with
distributed computing	Worth being familiar with
compiled languages	Worth being familiar with
OOP	Worth being familiar with
symbolic math	Worth being familiar with
data bases	Worth being familiar with
I/O	Worth being familiar with
Flash	Worth being familiar with
HTTP	Worth being familiar with
XML	Worth being familiar with
SOAP	Worth being familiar with
SVG	Worth being familiar with
KML	Worth being familiar with
grid	Worth being familiar with
lattice	Worth being familiar with
event programming	Worth being familiar with

maps	Worth being familiar with
interactivity	Worth being familiar with
animation	Worth being familiar with
perception	Worth being familiar with
color	Worth being familiar with
raster/vector graphics	Worth being familiar with

Potential benefits of incorporating backward design ideas include clearer articulation of goals and better assessment of goal achievement.

We see our framework as one contribution to scholarship on improving data science training programs. Given the increasing economic need, in the USA and abroad, for data scientists and other researchers with quantitative training, we anticipate that our framework and its future iterations will continue to prepare students for data science careers by offering training in tangible and transferable analytic skills in the context of solving scientific questions.

Integrating more mentoring activities

Our framework would benefit students more if we explicitly incorporate more mentoring activities. Through professional development courses at the university, we received training in how to offer professional support to students. While we both informally supported our students, the professional development program suggested ways to encourage the student's professional development through structured conversations and goal-setting. Additions like this would only enhance our framework.

Baker and Griffin (2010) discuss the role of faculty "developers" in student success. A faculty "developer", as envisioned by Higgins and Kram (2001), offers not only psychosocial and career support, like a mentor, but also supports students' academic goals. Such relationships between developers and students benefit both parties. The student gets support while the developer refines her teaching and expands her scholarly network. We anticipate expanding our framework to more holistically support students.

References

Baker, Vicki L, and Kimberly A Griffin. 2010. "Beyond Mentoring and Advising: Toward Understanding the Role of Faculty 'Developers' in Student Success." *About Campus* 14 (6): 2–8.

Higgins, Monica C, and Kathy E Kram. 2001. "Reconceptualizing Mentoring at Work: A Developmental

Network Perspective." *Academy of Management Review* 26 (2): 264–88.

Lin, Cindy Xide, Qiaozhu Mei, Jiawei Han, Yunliang Jiang, and Marina Danilevsky. 2011. "The Joint Inference of Topic Diffusion and Evolution in Social Communities." In *2011 Ieee 11th International Conference on Data Mining*, 378–87. IEEE.

Nolan, Deborah, and Duncan Temple Lang. 2010. "Computing in the Statistics Curricula." *The American Statistician* 64 (2): 97–107.

Pelled, Ayellet, Josephine Lukito, Fred Boehm, JungHwan Yang, and Dhavan Shah. 2018. "'Little Marco,'Lyin'Ted,'Crooked Hillary,' and the 'Biased' Media: How Trump Used Twitter to Attack and Organize." In *Digital Discussions*, 176–96. Routledge.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David. 2016. "Text Analysis of Trump's Tweets Confirms He Writes Only the (Angrier) Android Half." http://varianceexplained.org/r/trump-tweets/.

"Sampled Stream." n.d. https://developer.twitter.com/en/docs/labs/sampled-stream/overview.

Vance, Eric A, Erin Tanenbaum, Amarjot Kaur, Mark C Otto, and Richard Morris. 2017. "An Eight-Step Guide to Creating and Sustaining a Mentoring Program." *The American Statistician* 71 (1): 23–29.

Wells, Chris, Dhavan V Shah, Jon C Pevehouse, JungHwan Yang, Ayellet Pelled, Frederick Boehm, Josephine Lukito, Shreenita Ghosh, and Jessica L Schmidt. 2016. "How Trump Drove Coverage to the Nomination: Hybrid Media Campaigning." *Political Communication* 33 (4): 669–76.

Wiggins, Grant, and Jay McTighe. 2005. *Understanding by Design*.