




WEB701 PROJECT TWO – INTEGRATING R PROGRAMMING LANGUAGE IN WEB APPLICATION



Lionell Carlo Paquit
GRADUTE DIPLOMA in Information Technology

Table of Contents

Section 1. Introduction	1
Section 1.1. Twitter	1
Section 1.2. R Programming	1
Section 2. Project Implementation.....	2
Section 2.1. Comprehensive R archive network (CRAN)	2
Section 2.2. Pre-processing Tweets	2
Section 2.3. Integrating R Programming into Web Application using Shiny	3
Section 3. Problems Encountered	4
Section 4. Impact of the Technology	5
References	6

Section 1. Introduction

Over the years, the growth of Web 2.0 has been very explosive (Joshi and S, 2008). Several types of social media like blogs, discussion forums, review websites and community websites can be very useful to determine public opinion or sentiment towards a certain topic or brand. According to Alexa (2018), Twitter had become the world's 12th most popular website in June 2018. The website was launched in July 2006 and there was noticeable growth in popularity since then. Part of the reason is because celebrities tweeting regular updates about their daily lives (Johnson, 2009).

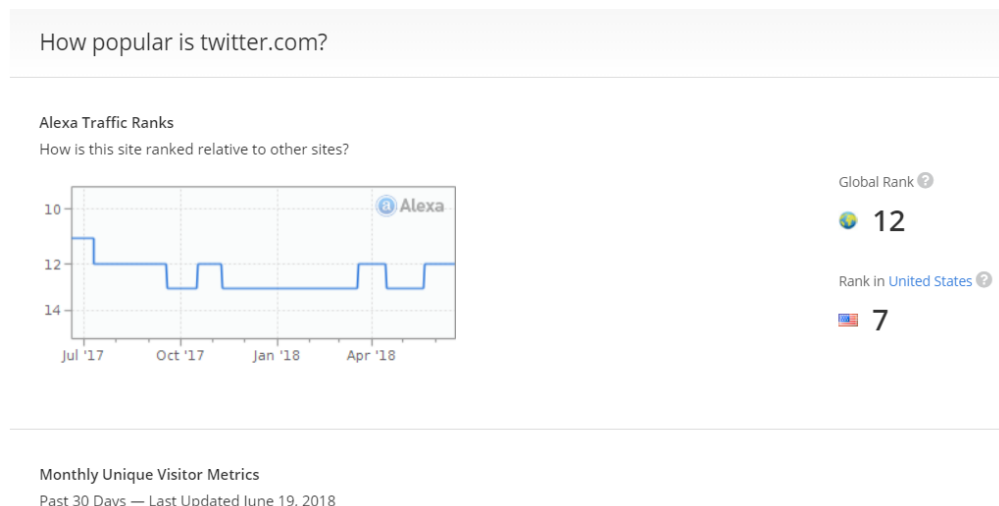


Fig. 1. Screenshot from Alexa Inc. of Twitter.com website traffic overview.

Section 1.1. Twitter

Twitter is a popular microblogging service that gain exponential growth over the recent years. It has millions of users including celebrities and media personalities. It allows users to create status messages called “tweets”. These tweets are limited to only 140 characters per posting allowing people to share easily quick updates and interest with brevity in their expressions (Educause Learning Initiative, 2007). Millions of tweets are written every day includes comments and conversation about current topics in the world events, as well as links to websites, blog posts, photos or videos. And sometimes these tweets expressed valuable contents like viewpoints and opinions on various topics. These resources have been identified by many organizations as a gold mine for marketing knowledge. One of the many ways in taking advantage of this enormous data on Twitter is by sentiment analysis.

Section 1.2. R Programming

Procedures and statistical analysis will be implemented in R programming language. R is a language and a free software environment for statistical computing and graphics. R is a powerful tool that provides a wide variety of statistical and graphical techniques and can be extended easily via packages. What sets it apart from any other statistical tools like SAS and IBM was that it is free and open source, and runs well on variety of UNIX platforms, Windows and MacOS. It contains advanced statistical routines as well as large, coherent and integrated collection of tools for data analysis. It also processes powerful graphics capabilities which aid

in visualization of data and results significantly. Because of these properties, more control on the program allowed user to interact with the application at optimal ease.

Section 2. Project Implementation

Section 2.1. Comprehensive R archive network (CRAN)

CRAN is a network file transfer protocol (ftp) and web servers around the world that store identical, up-to-date, versions of code and documentation for R. It acts as the download area, carrying the software itself, extension packages, PDF manuals; in short everything you need to download using R. The study needed a list of library packages to be able to implement data analysis in R:

- **tm package.** The tm package provides a comprehensive text mining framework for R. The tm package offers functionality for managing text documents. The package provides native support for reading in several classic file formats (e.g. plain text, CSV, or XML files). There is also a plug-in mechanism to handle additional file formats. Tm provides easy access to preprocessing and manipulation mechanisms such as whitespace removal, stemming, or stopword deletion. tm is freely available under the GNU General Public License (GPL) (Feinerer, Kurt Hornik, & Artifex Software, 2018).
- **Snowballc package.** The SnowBallC is an R interface to the C libstemmer library that implements Porter's word stemming algorithm for collapsing words to a common root to aid comparison of vocabulary (Bouchet-Valet, 2015).
- **twitterR package.** The twitter package provides an interface to the Twitter web API (Gentry, 2016).
- **wordcloud2 package.** A fast visualization tool for creating wordcloud by using 'wordcloud2.js'. 'wordcloud2.js' is a JavaScript library to create word presentation on 2D canvas or HTML (Lang, 2018).
- **ggplot2 package.** A system for 'declaratively' creating graphics, based on ``The Grammar of Graphics''. You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details (Wickham, Chang, & RStudio, 2016).
- **shiny package.** Shiny makes it easy to build interactive web applications with R. Automatic "reactive" binding between inputs and outputs and extensive prebuilt widgets make it possible to build beautiful, responsive, and powerful applications with minimal effort (Chang, Cheng, Allaire, Xie, & McPherson, 2018).

Section 2.2. Pre-processing Tweets

Pre-processing the tweet is the process of cleaning and preparing the tweet for classification. Tweets usually contain lots of noise and uninformative text that do not contribute to its analysis. Many tweets include URLs, tags to other users, or symbols that have no meaning. To accurately obtain a tweet's data, the noise that would not help in the analysis process was removed.

Several pre-processing steps were made to clean the tweets, and these include:

1. Remove ASCII characters from text

2. Removing URLs. It is common to find URLs in tweets. All strings that describe links or hyperlinks were removed.
3. Remove numbers. Numbers in general, do not represent a sentiment thus, numbers were removed in tweets.
4. Tokenization. Tweets were split up into terms or tokens by spaces forming a list of individual words per tweet.
5. Case normalization. The entire data were changed to lower case. The reason for this was because the classifier will treat words like, amazing, Amazing, amAZinG and AMAZING as different words. This method greatly reduced the sparsity of our data set.
6. Removing punctuations. All punctuations and other symbols were removed as they add no value to the text.
7. Strip multiple whitespaces. Strip extra whitespace from a tweet and characters were collapsed to a single space.
8. Removing stop words. Stop words are words that do not add meaningful content to the tweet. Removing them reduced the space of the items significantly in the training and testing set. Example of such words include, *the, a, and, to, of*, etc.
9. Remove sparse terms. Terms that appear very often were removed. This pre-processing step was necessary to reduce computational cost because the more the terms means it takes longer to build the classifier.
10. Stemming. It is the process of removing prefix and suffix leaving the stem or the root of the word. For example, the set of words *read, reader, readers*, and *reading* all will be reduced to the root word *read*. SnowBallC stemmer which is a package in R was used to implement this process.

Section 2.3. Integrating R Programming into Web Application using Shiny

In order to publish our R application using Shiny package, we will need to create account in shinyapps.io. Shinyapps.io is a platform as a service (PaaS) for hosting Shiny web applications (apps). Once you have set up your account in shinyapps.io, retrieve your token which is automatically generated from the shinyapps.io. Select the Tokens option in the dashboard menu.

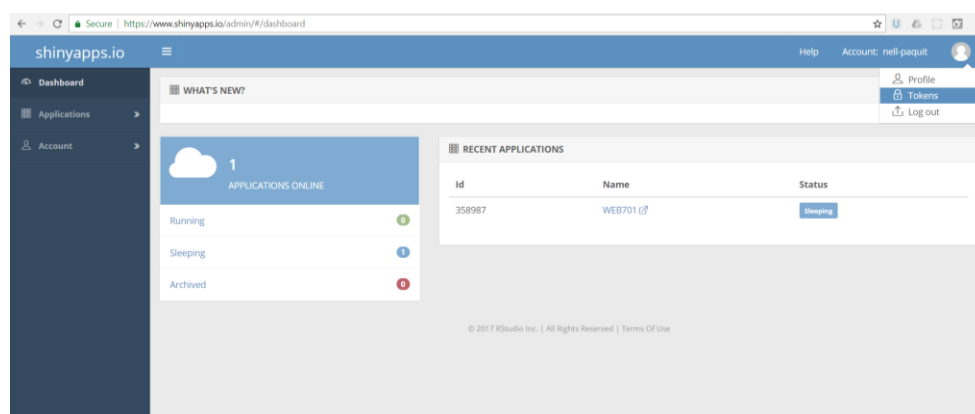


Fig. 2. Screenshot from shinyapps.io dashboard.

When your token is setup, you can manage your shinyapps.io account by going to Tools -> Global Options -> Publishing. Then you can just publish you shinny application by simply going to RStudio>File>Publish and a window will popup.

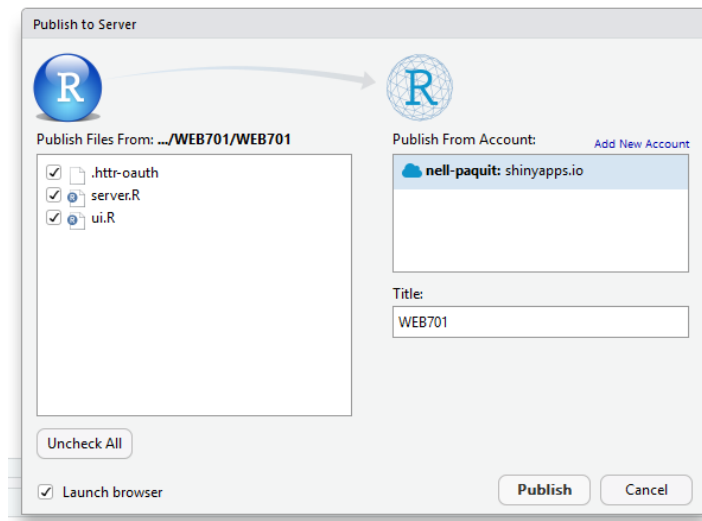


Fig. 3. Publish R application in shinyapps.io account.

Every application you deploy will have a unique URL, served over a secure socket (SSL) connection and accessible from a web browser in the format:

```
https://<accountname>.shinyapps.io/<applicationname>
```

You can embed your application within other pages by using an `iframe`. Here is an example of an `iframe` for a fictitious application. Note that you may want to size the frame differently based on your application's display requirements.

```
<iframe id="example1" src="https://<accountname>.shinyapps.io/<applicationname>"
style="border: none; width: 100%; height: 500px"
frameborder="0">
</iframe>
```

Here are the links of my shiny apps and the webpage I embed the apps:

<https://nell-paquit.shinyapps.io/WEB701/>

<http://newsimland.com/~nell/shinny-apps-web-page-intergration/>

Section 3. Problems Encountered

There are several problems and limitations I encountered while making the applications or coming up with solutions of integrating R into web.

Applications deployed on shinyapps.io are accessible by loading a URL of the form: `https://<account-name>.shinyapps.io/<application-name>/`. If we would like to have greater control over the URLs that their applications are served on, we can subscribe to the Professional plan and host the application on domains that belongs to us. In addition with free plan, you can only run one instance of your application, meaning if you wanted your

application to run in different places simultaneously you needed the premium plan for it. The bundle size that can be uploaded is limited to 1 GB for the Free plans which means we are allowed to use 1024MB of memory. Since it is deployed on external server we will not have guarantees regarding the number of CPU cores, or the speed of the CPUs that are allocated to the deployed applications. Thus if our application is computationally extensive we might experience slow process or loading.

The maximum number of instances you can add is governed by your subscription plan:

Plan	Instances
Free	1
Starter	1
Basic	3
Standard	5
Professional	10

Instance	Memory
small	256 MB
medium	512 MB
large (default)	1024 MB
xlarge	2048 MB
xxlarge	4096 MB
xxxlarge	8192 MB

Fig. 4. Rate limits in shinyapps.io accounts plans.

There are also limitations on using Twitter API with free accounts in terms of the number of request daily or per-user basis. Sources of this limitations can be viewed through this links:

[Rate Limits](#) - Standard API Rate limits per window

[Twitter Rate Limiting](#)

Section 4. Impact of the Technology

Not only since the controversy of Cambridge Analytica with Facebook data, social sites data like Facebook and Twitter have already been used in various studies like marketing and social research. Data gives important insights that offer and guide an organization in decision making. Government, for example, has been using this technology to get the sentiment of people when proposing a new legislation (Zavattaro, French, & Mohanty, 2015). Data has also been commonly used as a tool in marketing research.

With this technology, we will explore on how this data will be presented in a manner where it is accessible to everyone. And one of the best way in presenting this data is through the World Wide Web. Although there have been other software or cloud computing tool that has been used already, these tools are expensive and very complicated to use for small

organization. Using R with simple instructions and powerful visualization of data is found to be very useful even though it is not recommended for large datasets like those big organization uses.

By and large, R is very useful for small organization and students who wanted to learn data analytics. And using packages like Shiny to integrate it to the Web will greatly help them in sharing their reports and even their program much more easily. Aside from the fact that is a free and open source platform, it provides powerful statistical and graphical representation of data analytics. It will be interesting to know what this tool can do in the future where data is vast and will become more important than money.

References

- Bouchet-Valet, M. (2015). Snowball stemmers based on the C libstemmer UTF-8 library, 1–5.
- Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2018). Package “shiny”. Web Application Framework for R. *CRAN R Project*.
- Educause Learning Initiative. (2007). 7 Things You Should Know About Twitter. Retrieved June 12, 2018, from <https://library.educause.edu/~media/files/library/2007/7/eli7027-pdf.pdf>
- Feinerer, I., Kurt Hornik, & Artifex Software, I. (2018). Package ‘tm.’ Retrieved from <https://cran.r-project.org/web/packages/tm/tm.pdf>
- Gentry, J. (2016). Package ‘twitter,’ 32. Retrieved from <http://lists.hexdump.org/listinfo.cgi/twitter-users-hexdump.org>
- Lang, D. (2018). Package ‘wordcloud2.’ Retrieved from <https://cran.r-project.org/web/packages/wordcloud2/wordcloud2.pdf>
- Wickham, H., Chang, W., & RStudio. (2016). Ggplot2. Retrieved from <https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf>
- Zavattaro, S. M., French, P. E., & Mohanty, S. D. (2015). A sentiment analysis of U.S. local government tweets: The connection between tone and citizen involvement. *Government Information Quarterly*, 32(3), 333–341. <https://doi.org/10.1016/j.giq.2015.03.003>