

Relp

Taikgun Song

Due May 5th, 2017

Abstract

A popular webpage `yelp.com` was reconstructed into a `shiny` application. The application focuses on imitating two basic features of the webpage: providing user-generated reviews/ratings, and providing basic information of the local businesses. The imitation was done by first scraping `yelp.com` webpage, and re-building the features with interactive visualization and data re-organization via `Shiny` package built in `R`.

Introduction

`Relp`(reads Re-lp) stands for re-construction of `yelp.com` using `R`, and the meaning of the name pretty much describes what the application is already. The motivation for this project was to learn about the `Shiny` (Chang and others 2017) application - a web application framework for `R`(R Core Team 2017), and to fulfill my personal interest - constructing a web page, and understand how HTML language works. The idea is to scrape certain information from `yelp.com` using the `rvest`(Wickham and RStudio 2016) package and build a web application using `R` packages such as `tidyverse`(Wickham and RStudio 2017), `leaflet`(Cheng 2017), `ggmap`(Kahle and Wickham 2016), `DT`(Xie 2016), and `shiny`.

Basic features of `yelp.com`

`yelp.com` (“Yelp” 2017) is an online and mobile application providing platforms for internet users to publish reviews and rate local businesses, as well as a platform for online delivery service and reservation service. For the project, only the user-generated reviews and local business information was taken into consideration. The front page of `yelp.com` is illustrated in Figure 1. The front page of `Relp` application will have a similar layout the the below figure with images and widgets. Users of `yelp.com` can gain information of the local businesses by entering the location information and topic of interest.



Figure 1: Front page of ‘yelp.com’

Once the above information are entered, the page is then redirected to Figure 2a. The main page has many features, but only a few of them is reflected in `Relp`: filter by price, map, and a list of businesses. When the cursor is hovered over a business, marker corresponding to the business changes its color. This feature is also realized in `Relp`.

When a single business is selected from the list, the page is redirected to Figure 2b. This page contains detailed information of the business such as its phone number, hours of operation, more business info, images, and reviews/rates written by the users. The mentioned information are scraped, organized, and displayed on Relp.

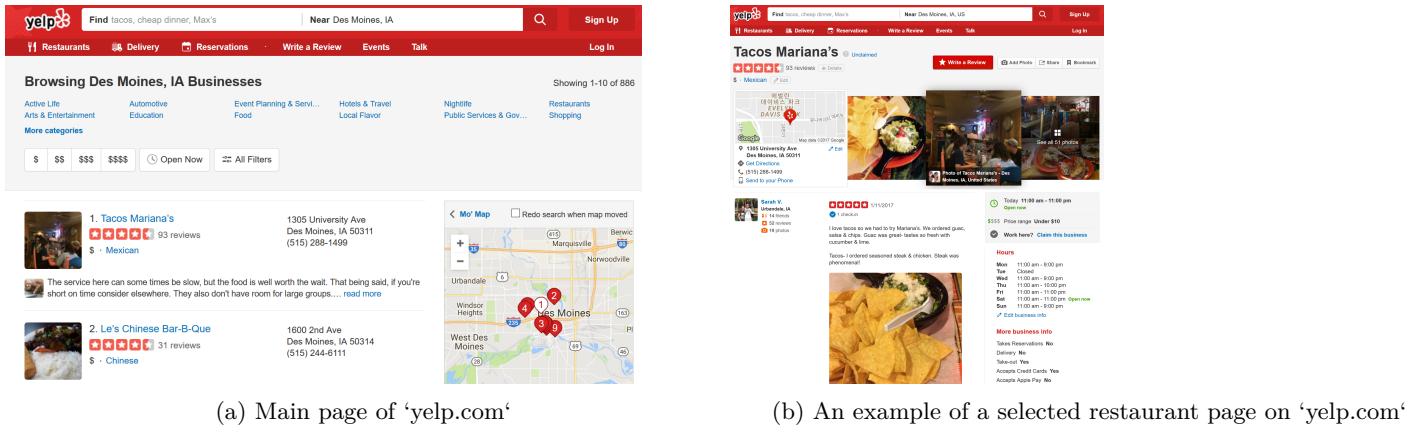


Figure 2: Example of 'yelp.com'

Relp

In order to replicate yelp.com, Relp has four separated procedures: variable input, web scraping, data manipulation/organizing, and data display. Widget or leaflet and DT output is used as variable inputs. Once the values of the variables are selected, rvest is used to scrape yelp.com. xpath expression is used to select node(s). Then, tidyverse, ggmap and some base R function is used to manipulate/organize the scraped data. Finally, leaflet, DT, base R function, and HTML expressions are utilized to display the output. Flowchart of Relp is provided in Figure 3 for a better understanding.

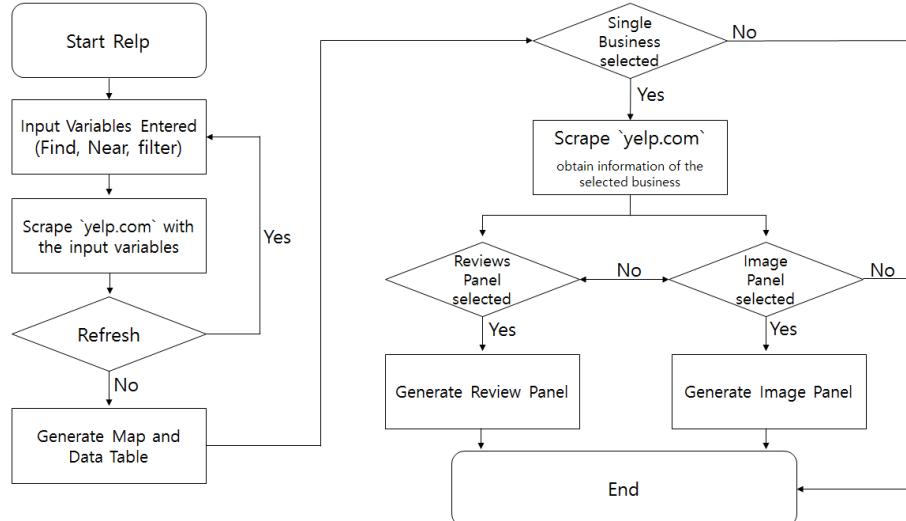


Figure 3: Flowchart of Relp application

Front panel

The front page of the **Relp** tried to mimic the front page of [yelp.com](https://www.yelp.com). As illustrated in Figure 4, the page has an image of a delicious dish and a couple of widgets to enter inputs for the application. When the search button is clicked on the real [yelp](https://www.yelp.com) front page, the user is directed to a new web page with a list of businesses information. The realization of this feature was constructed on **Relp** by using the conditional panel with the “search” action button for initiation. That is, when the red search icon is clicked with appropriate “Find” and “Near” variable, the shiny application switches its panel to the main panel. The search URL of [yelp.com](https://www.yelp.com) has a general form of https://www.yelp.com/search?find_desc=Find&find_loc=Near. Simple `sub` function can change the “Find” and “Near” variable to a widget input. The application starts scraping information from [yelp.com](https://www.yelp.com) once the search button is pressed. Location information is then translated to longitude and latitude via `ggmap` package.

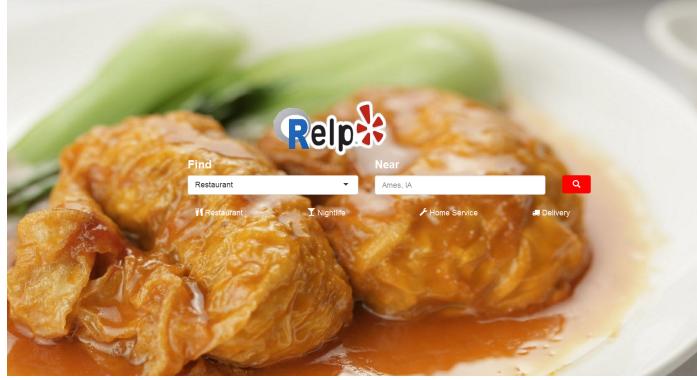


Figure 4: Front page of the Relp application

Main panel

The main panel of the **Relp** application consist of two different sections: the input section and the output section. The input section has multiple widgets and is located on the left column of the page. User of **Relp** application could change the “Find” and the “Near” variable, and also filter the price variable. New output could be generated by clicking the “refresh” button. The output section located on the right side of the page with the following three sub-panels: map, review, and image panels. An example of the main page is provided in Figure 5. The “Find” and the “Near” variable was changed from “Ames, restaurant” in Figure 5a to “San Francisco, Bars” in Figure 5b.



(a) Initial map panel of Relp application

(b) Map panel of Relp application with different variables

Figure 5: Example of the use of widget on the main page

Map

Location, price, and average rating are important factors when choosing the right business. These information could be obtain in the map panel. Visualization of the location information is reflected as a map using the `leaflet` package. Other information are organized into a data table by the `DT` package. Interaction between the data table and the map is also established. As illustrated in Figure 5b, when a row in the data table is highlighted, the corresponding marker on the map changes its color from blue to red.

Review

Crowd-sourced reviews and ratings often provide valuable indirect experiences of a business to the users reading the evaluation. The review panel displays phone number, hours of operation, and features of a particular business as well as the crowd-sourced reviews and ratings. An example of the review panel is given in Figure 6. When a row of the data table from the map panel is highlighted, the corresponding URL of the business is retrieved. The web page of the URL is then scraped and re-organized as a shiny web panel.

The screenshot shows the Relp application's Review panel for 'Aunt Maude's'. At the top, there are tabs for Applications, Map, Reviews, and Photos. The Reviews tab is selected. The search bar shows 'Near Ames, IA' and the category 'Find Restaurant'. Below the search bar are filters for 'Choose Price:' with options: \$ Inexpensive (checked), \$\$ Moderate, \$\$\$ Pricey, and \$\$\$\$ Ultra High-End. The main content area displays a list of reviews with columns for 'star', 'posted', 'text', and 'More Business Info'. Each review includes a timestamp and a short summary. For example, a 5.0 rating from 1/10/2016 says: 'The further West are an absolute MUST here. Had a light dinner and just had a salad for dinner but the service was excellent. Our server was ready on her A-game and was very friendly with how professionally she approached every interaction. I would highly recommend this place if you're in the area.' To the right of the reviews, there is a sidebar with 'Phone (515) 233-4136' and 'Price Range \$11-30'. Below that is a 'Hours' section showing availability from Monday to Sunday. At the bottom right is a 'More Business Info' section with various business details like 'Dishes Vegetarian' and 'Accepts Credit Cards'.

Figure 6: Review panel of Relp application

Images

Pictures of a business is also an important factor when making decisions. The algorithm for generating the image panel is similar to that of the review panel. Image files of a business are first download in a temporary folder. Two action buttons are used to navigate the images. When a new business is selected, files in the temporary folder are deleted and new images are downloaded. Display of the image panel is portrayed in Figure 7.

The screenshot shows the Relp application's Image panel for 'Aunt Maude's'. The interface is similar to the Review panel, with tabs for Applications, Map, Reviews, and Photos. The Photos tab is selected. The search bar shows 'Near Ames, IA' and the category 'Find Restaurant'. Below the search bar are filters for 'Choose Price:' with options: \$ Inexpensive (checked), \$\$ Moderate, \$\$\$ Pricey, and \$\$\$\$ Ultra High-End. The main content area shows a large thumbnail image of the restaurant's exterior. Navigation buttons '< Previous' and '> Next' are visible on either side of the image. The image itself shows a brick building with a sign that reads 'MAUDE'S'.

Figure 7: Image panel of Relp application

Future work

Although much effort was put into the application to imitate that yelp.com, the application is not perfect. There are a few known bugs to be fixed, some yelp.com features are omitted, and the cosmetic part may be improved for better display.

Fix bugs

First, the application will crash when the Review and Image panel is selected with more than one business. This is because the script is written for only one selected business URL. A simple statement (e.g. the `if` statement) may be applied to check if more than two business URLs are chosen as an input, and generate a warning message when the condition is not met.

Second, a script should be updated to un-mark the selected marker on the leaflet output when clicked twice. Current script only marks the chosen input, and no code is written to reverse the selection. A leaflet proxy code could be applied to remove the marker when the business is clicked again.

Third, an error occurs when publishing the application onto the `shinyapps.io`. On the local machine, the application runs smooth without any issues regarding the `leaflet` package. However, uploading the application yields the following message: `ERROR: could not find function "leafletOutput"`. Apparently, similar issues were brought up a couple years ago (2015). One of the suggestion was to load the `leaflet` package on both `ui` and `server`, however, this is not a solution for `Relp` case. Other possible source of errors maybe due to the lack of version control or errors generated from `leafletProxy` and `htmlwidgets`. Further investigation should take place to fix this problem.

Add more features

One simple, but effective feature that could be easily implemented in `Relp` is the navigation feature. A new text input widget may be introduced to enter the current location as “From”, and the location information of the of the business as “To”. `ggmap` has a function that calculates route from a location to another location. This route can be displayed using `ggmap` or `leaflet`.

Improve speed

Slow speed of `Relp` is indeed an important problem to be solved. The fundamental reason of this problem is due to web scraping (especially fetching). A pseudo-treatment for this problem was suggested in the application by presenting a conditional panel, as shown in Figure 8, that may entertain the users while waiting for the result to appear.

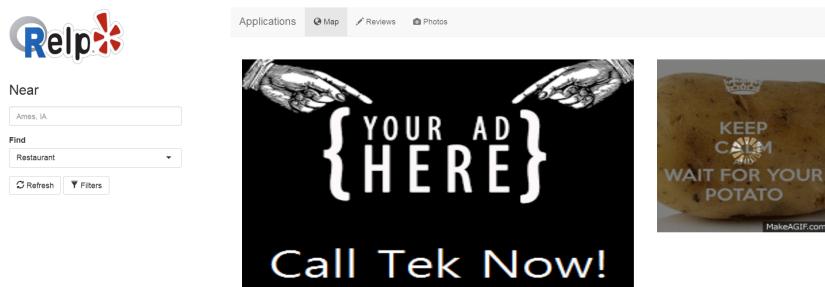


Figure 8: Conditional panel when `Relp` application is "busy"

Of course, this “pseudo” treatment is not a practical solution. In order to truly solve this problem, `Relp` must have a local database of its own. A suggestion is to write a bot or a web crawler script that consistently scrapes `yelp.com` data into a local database. Loading data from a local repository will drastically increase the speed of `Relp`.

Beautification

As mentioned in the introduction, I have great interest in HTML/css language. If more time was given, I would have entertained myself incorporating HTML and css modification to `Relp` so that the appearance of `Relp` is similar to those commercial web pages.

Conclusion

Although there are minor bugs to be fixed and future improvement to be made, `Relp` does what it is supposed to do: scrape data from `yelp.com` and rebuild plots, tables, and post photo images. While building `Relp` materials and packages introduced in STAT585 was very helpful. Next challenge is to successfully publish `Relp` on `shinyapps.io` and fix known bugs.

References

- Chang, Winston, and others. 2017. *Web Application Framework for R*. <https://cran.r-project.org/web/packages/shiny/shiny.pdf>.
- Cheng, Joe. 2017. *Create Interactive Web Maps with the Javascript 'Leaflet' Library*. <https://cran.r-project.org/web/packages/leaflet/leaflet.pdf>.
- Kahle, David, and Hadley Wickham. 2016. *Spatial Visualization with Ggplot2*. <https://cran.r-project.org/web/packages/ggmap/ggmap.pdf>.
- R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <http://www.R-project.org/>.
- Wickham, Hadley, and RStudio. 2016. *Easily Harvest (Scrape) Web Pages*. <https://cran.r-project.org/web/packages/rvest/rvest.pdf>.
- . 2017. *Easily Install and Load 'Tidyverse' Package*. <https://cran.r-project.org/web/packages/tidyverse/tidyverse.pdf>.
- Xie, Yihui. 2016. *A Wrapper of the Javascript Library 'Datatable'*. <https://cran.r-project.org/web/packages/DT/DT.pdf>.
- “Yelp.” 2017. <https://www.yelp.com/>.