Data Wrangling

Sine Gov

December 30, 2017

Data Sources

The data for this project was gathered from three sources: Reddit, the Official Overwatch Forums, and Twitter. Since we are performing sentiment analysis For this project, we are only interested in collecting the text comments from each of these sites. Each section below outlines how data was gathered and cleaned for each site.

Twitter

Using the twitteR and ROAuth libraries, the hashtag #Overwatch was used to collect tweets. After authenticating the session with Twitter, unique tweets in English were targeted for collection. Since the twitteR package outputs the data in a clean form, no additional data wrangling was needed.

Code to scrape tweets from twitter

```
...

# Grab latest tweets
print("Grabbing tweets!!!!! Will take a second.")
tweets_OW <- searchTwitter("#overwatch, -filter:retweets", n=14586, lang = "en")

df_tweets_OW <- twListToDF(tweets_OW)
write.csv(df_tweets_OW, file = "OWT_12-22.csv")
```

Tweets scraped

```
## Loading required package: bitops
```

X.U.FEFF.	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	id	replyToUID	statusSource	screenName	retweetCount	isRetweet	retweeted	longitude	latitude
1	Merry (early) Christmas to me. #overwatch @PlayOverwatch https: //t.co/LZRLBOEylQ	FALSE	0	NA	12/22/2017 19:36	FALSE	NA	9.44290e+17	NA	Twitter for iPhone	realBOBtheHOB	0	FALSE	FALSE	NA	NA
2	Going live in 10minutes!! Starting off the holidays with some video games! Come join me? What will we play tonight?… https: //t.co/BtjNJtzrEL	FALSE	0	NA	12/22/2017 19:35	TRUE	NA	9.44290e+17	NA	TweetDeck	twistyoc	0	FALSE	FALSE	NA	NA

X.U.FEFF.	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	id	replyToUID	statusSource	screenName	retweetCount	isRetweet	retweeted	longitude	latitude
3	On the set w/ @mrdubbsie @bhanford #gwne #gwnenow #esports #overwatch #esportsfrance… https://t.co/6Vh6cC2utT	FALSE	0	NA	12/22/2017 19:35	FALSE	NA	9.4429e+17	NA	Instagram	heygirlhey	0	FALSE	FALSE	-118.243	34.0522
4	Slid back into gold…..SR quicksand?? #Overwatch #OverwatchWinterWonderland https://t.co/ToVg2WNYOv https://t.co/JPGLSQXQSJ	FALSE	0	NA	12/22/2017 19:34	FALSE	NA	9.4429e+17	NA	Twitter Web Client	aimbothero51	0	FALSE	FALSE	NA	NA
5	https://t.co/A0OFHcuUqJ come watch me play #overwatch #OverwatchWinterWonderland #stream #twitch #smallstreamers… https://t.co/CXGuUGF5pW	FALSE	0	NA	12/22/2017 19:30	TRUE	NA	9.4428e+17	NA	Twitter for iPhone	Forsythiagrace	1	FALSE	FALSE	NA	NA

Reddit

A custom script was written to gather data from reddit. The script first gathers all of the hyperlinks of all the topics on the first page of the subreddit /r/Overwatch. Once it has all the necessary links, it visits each link and copies each comment, the date of the comment, and the user name. The libraries listed in the code below were used to scrape this website.

Code to scrape hyperlinks of topics on the front page

```r
library(rvest)
library(tidyr)
library(RCurl)
library(dplyr)
library(curl)

#Read URL
ow.forums <- read_html("https://www.reddit.com/r/Overwatch/")

#scrapes links from read URL
links <- ow.forums %>% html_nodes("a") %>% html_attr("href")

#Makes scrapped links a data frame
```

```r
links <- as.data.frame(links, row.names = NULL)

#Find the links for the forums (since links are /forums) and keep these.
keep<- grepl("^https://www.reddit.com/r/Overwatch/comments/", links$links)
#print(keep)
keep <- as.data.frame(keep, row.names = NULL)

#bind the answers to the new dataframe
links <- cbind(links, keep)

#remove the rows of unneeded links
#links now contains all of the links for the forum post to be scrapped
#paste to strip out unnecessary rows
links <- links[links$keep == "TRUE",]
links <- paste(links$links)

#new df ow.total
owreddit.total <- data.frame("link", "title", "time", "text", "user", stringsAsFactors = FALSE)
print(owreddit.total)
```

Code to scrape data from each topic

```r
#While loop iterates through scraped links for comments.
n <- 1
while(n <= length(links)) {
  read_single_links <- read_html(links[n])

  #scrapes the link's title
  link_title <- gsub("https://www.reddit.com/r/Overwatch/comments/", "", links)
  link_title <- substr(link_title,8, nchar(link_title)-1)
  link_title <- gsub("_", " ", link_title)

  #scrapes comments
  comments <- read_single_links %>% html_nodes(".md")

  #convert from xml to usable format
  comments <- trimws(comments)
  comments <- gsub("<.*?>", "", comments)

  #Cleans comments by replacing \n html code
  comments <- gsub("\n", "", comments)

  #scrapes the .tagline node which has username data
  reddit_user <- read_single_links %>% html_nodes(".tagline")

  #Scrapes to the user name
  reddit_user <- gsub('.*user/', "", reddit_user)

  #Scrapes the stuff after class out
  reddit_user <- gsub('class.*', "", reddit_user)

  #clean up of user name
  reddit_user <- gsub(" ","",reddit_user, fixed=TRUE)
  reddit_user <- gsub("\"","",reddit_user, fixed=TRUE)
  reddit_user <- gsub("<p","",reddit_user, fixed=TRUE)

  #scrapes comment timestamp by date and cleans it
  comment_time <- read_single_links %>% html_nodes(".tagline, time")
  comment_time <- gsub('.*datetime=', "", comment_time)
  comment_time <- gsub('T.*', "", comment_time)
  comment_time <- gsub('\"', "", comment_time)
  comment_time <- gsub('<p.*', "", comment_time)
```

Code to write all the scraped data into a csv file

```
#adds topic to a vector with link, title using a for loop
  for(i in 1:length(comments)) {
    owreddit.total <- rbind(owreddit.total, c(links[n], link_title[n], comment_time[i], comments[i+2], i
  }

  print(n)
  n<-n+1
  #Sys.sleep(2)
}

write.csv(owreddit.total, file = "reddit_comments.csv")
```

Overwatch subreddit scraped

	X.link.	X
2	https://www.reddit.com/r/Overwatch/comments/7jd5bw/its_the_most_wonderful_time_of_the_year_and_we/	it
3	https://www.reddit.com/r/Overwatch/comments/7jd5bw/its_the_most_wonderful_time_of_the_year_and_we/	it
4	https://www.reddit.com/r/Overwatch/comments/7jd5bw/its_the_most_wonderful_time_of_the_year_and_we/	it
5	https://www.reddit.com/r/Overwatch/comments/7jd5bw/its_the_most_wonderful_time_of_the_year_and_we/	it
6	https://www.reddit.com/r/Overwatch/comments/7jd5bw/its_the_most_wonderful_time_of_the_year_and_we/	it

Overwatch Forums

Simliar to the reddit, a custom script was written to gather data from the official Overwatch forums. The script first gathers all of the hyperlinks of all the topics on the first page of the subreddit /r/Overwatch. Once it has all the necessary links, it visits each link and copies each comment and and the user name. Note that there is code to scrape date data, however it works in rare instances. Since date is not necessary for the analysis later, no further effort was made to clean this up and will be removed in the final version of this script. The libraries listed in the code below were used to scrape this website.

Code for gathering links

```
library(rvest)
library(stringr)
library(tidyr)
```

```r
library(RCurl)
library(dplyr)
library(curl)


#Read URL
ow.forums <- read_html("https://us.battle.net/forums/en/overwatch/22813879/")

#scrapes links from read URL
links <- ow.forums %>% html_nodes("a") %>% html_attr("href")

#Makes links scrape a data frame
links <- as.data.frame(links, row.names = NULL)

#Find the links for the forums (since links are /forums) and keep these.
keep<- grepl("^/forums", links$links)
keep <- as.data.frame(keep, row.names = NULL)

#bind the answers to the new dataframe
links <- cbind(links, keep)

#remove the rows of unneeded links
#links now contains all of the links for the forum post to be scrapped
links <- links[links$keep == "TRUE",]

#links only contains the tail end of the links. creating front_url to be concatenated later
front_url <- "https://us.battle.net"


#concatenate to create a full link and remove duplicates
full_links <- paste(front_url,links$links, sep = "")
#remove duplicate full_links
clean_links <- unique(full_links)


#new df ow.total
ow.total <- data.frame("link", "title", "time", "text", stringsAsFactors = FALSE)
```

Code for scraping data from each link and writing it to file

```r
#For loop to access URL from full_links. Scrapes titles and topic

n <- 1
while(n <= length(clean_links)) {
  read_single_links <- read_html(clean_links[n])

  #scrapes the title of the thread
  title_results <- read_single_links %>% html_nodes(".Topic-title")
  title <- xml_contents(title_results) %>% html_text(trim = TRUE)
  #print link and title below. line for to see new threads easier
  print("------------------------------------------------------------------------------")
  print(clean_links[n])
  print(title)

  #pulls content of thread
  topic_results <- read_single_links %>% html_nodes(".TopicPost-bodyContent")
  topic <- trimws(topic_results)

  #cleanup of bodycontent text.
  clean_topic <- gsub("<.*?>", "", topic)
  clean_topic <- gsub("\n", "", clean_topic)


  #pulls date
  topic_time <- read_single_links %>% html_node(".TopicPost-timestamp")
  clean_time <- gsub('<.*content=\\"', "", topic_time)
  clean_time <- gsub('\">.*', "", clean_time)
  clean_time <- as.POSIXct(clean_time, format = "%m/%d/%Y %I:%M %p")
  clean_time <- paste(clean_time)
  print(clean_time)

  #adds topic to a vector with link, title using a for loop
  for(i in 1:length(clean_topic)) {
    ow.total <- rbind(ow.total, c(clean_links[n], title, clean_time[i], clean_topic[i]))
  }

  print(n)
  n<-n+1

}
```

Additional Cleanup of Overwatch Forum Data

```
#A problem was found in the text in that some of it contained dates in the comment. Additional code was

####-----Cleaning up the Text column since it has dates merged into some of them
newdate <- c()
newtext <- c()

#for loop that looks for the date text andd removes it
for(i in 1:length(ow.total$X.text.)) {
  if(grepl('\\d\\d/\\d\\d/.*?M', ow.total$X.text.[i]) == FALSE) {
    print("CLEAN")
  } else {
    print("UNCLEAN! CLEANING")
  m <- regexpr('\\d\\d/\\d\\d/.*?M',ow.total$X.text.[i])
  owfdates <- regmatches(ow.total$X.text.[i], m, invert = FALSE)
  owcleantext <- gsub('\\d\\d/\\d\\d/.*?M', '', ow.total$X.text.[i])
  newdate[i] <- owfdates
  newtext[i] <- owcleantext
  }
}

#this loops replaces only the comments with date information with the cleaned up version
```

```
for(i in 1:length(ow.total$X.text.)) {
  if(is.na(newtext[i])) {
    print("Do not write over old value")
  } else if(newtext[i] != 'NA') {
    #print("Copy over new data")
    ow.total$X.text.[i] <- newtext[i]
    ow.total$X.time.[i] <- newdate[i]
  }
}

#remove duplicate values
str(ow.total)
newow.total_clean <- ow.total[!duplicated(ow.total[4]),]

#write the file
write.csv(newow.total_clean, 'OWFORUMS_manualcheck_12_30.csv')
```

Table for Overwatch Forum Data

```
library(knitr)
library(bitops)
#code to load in table to Rmarkdown
owforumtable <- read.csv(text=getURL("https://raw.githubusercontent.com/ujumqin/SB_CapstoneProject/maste

#code to display table
kable(owforumtable[6:10,])
```

	X.U.FEFF.link	title
6	https://us.battle.net/forums/en/overwatch/topic/20752619458	How do I know if someone I reported gets banned?
7	https://us.battle.net/forums/en/overwatch/topic/20752619458	How do I know if someone I reported gets banned?
8	https://us.battle.net/forums/en/overwatch/topic/20752619458	How do I know if someone I reported gets banned?
9	https://us.battle.net/forums/en/overwatch/topic/20752616810	McCree's Race/Nationality?
10	https://us.battle.net/forums/en/overwatch/topic/20752616810	McCree's Race/Nationality?