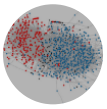
Finding Social Media Accounts, Automatically, for A List of Organizations.



CuriosityBits Data Lab

Feb 14, 2017 · 3 min read

In a recent post, I introduced using R to grab the links to over 8000 organizations' websites and social media accounts. The logic behind the web scrapping is simple. We use each organization name as the search term and extract the link of the top Google search result, assuming that the top link is the organization's official website. We do the same to get their social media sites by limiting the search to specific domains (i.e., Facebook.com and Twitter.com).

	OrganizationName	Issue.Area	Website	Twitter	Facebook
1	20/20 Vision	Environment; International Relations	www.aoa.org/patients-and-public/eye-and-vision...v...	https://twitter.com/metasmotivation	https://www.facebook.com/2020VisionClinic/
2	60 Plus Association	Senior Citizens; Social Security and Medicare	https://en.wikipedia.org/wiki/60_Plus_Association	https://twitter.com/60plusassoc	https://www.facebook.com/60PlusAssociation/
3	9to5, National Association of Working Women	Business, Labor, and Economics; Women and Men	www.9to5.org/	https://twitter.com/9to5org	https://www.facebook.com/9to5org/
4	AARP	Senior Citizens	join.aarp.org/	https://twitter.com/aarp	https://www.facebook.com/AARP/
5	Acton Institute for the Study of Religion and Liberty	Christian	www.acton.org/	https://twitter.com/actoninstitute	https://www.facebook.com/.../Acton-Institute-for-th.
6	Advocates for Highway and Auto Safety	Consumers; Health	saferoads.org/	https://twitter.com/saferoadsnow	https://www.facebook.com/.../Advocates-for-Highw..
7	Advocates for Youth	Children and Family	www.advocatesforyouth.org/	https://twitter.com/advocatestweets	https://www.facebook.com/Advocates4Youth/
8	AIDS Alliance for Children, Youth, and Families	Children and Family	www.washingtonpeacecenter.org/node/4820	https://twitter.com/aidsalliance_dc	https://www.facebook.com/AIDS-Alliance-for-Childr..
9	Alliance for Justice	Civil Rights; Crime and Justice	www.afj.org/	https://twitter.com/afjustice	https://www.facebook.com/AllianceforJustice/
10	Alliance for Retired Americans	Senior Citizens	https://retiredamericans.org/	https://twitter.com/activeretirees	https://www.facebook.com/retiredamericans/
11	American Association of Birth Centers	Health; Women's Health	www.midwife.org/American-Association-of-Birth-Ce...	https://twitter.com/nabccinfo/status/744857615453...	https://www.facebook.com/birthcenters/
12	American Association of People with Disabilities	Disabled	www.aapd.com/	https://twitter.com/aapd	https://www.facebook.com/DisabilityPowered/
13	American Association of University Women	Education; Women and Men	www.aauw.org/	https://twitter.com/aauw	https://www.facebook.com/AAUW.National/
14	American Cancer Society	Cancer; Health	donate.cancer.org/	https://twitter.com/americancancer	https://www.facebook.com/AmericanCancerSociety/
15	American Civil Liberties Union	Civil Rights	www.aclu.org/Donate	https://twitter.com/aclu	https://lm.facebook.com/l.php?u...aclu.org%2F&h...s=
16	American Civil Rights Institute	Civil Rights	acri.org/	https://twitter.com/wardconnerly	https://www.facebook.com/AmericanCivilRightsInstit..
17	American Coalition for Fathers and Children	Children and Family; Women and Men	www.acfc.org/	https://twitter.com/ffsjstockton	https://www.facebook.com/ACFCorg/
18	American Conservative Union	Conservative	conservative.org/	https://twitter.com/acuconservative	https://www.facebook.com/ACUConservative/

The output

In this post, I will introduce a way to cross-check whether the social media sites returned from the search are indeed the ones operated by the listed organizations. In doing so, we calculate the *match ratio*—the proportion of words in an organization's name that co-occur in the account page name returned from Twitter/Facebook API.

Step 1: Extract Twitter/Facebook handles from search result links.

For instance, we extract **AARP** from the link (**https://www.facebook.com/AARP/**). Here is the code that does the job:

First, let's load necessary libraries and the dataset. In my example, the data frame is named *d*, imported from a CSV file. We need to create two new columns to store the profile information from Twitter and Facebook API (named *NameFromTwitterAPI* and *NameFromFacebookAPI*, respectively).

```
library(Rfacebook)
library(twitteR)
library(stringdist)
library(stringr)
```

```
d <- read.csv("Orgs with social media accounts (web
scripting).csv")
```

```
d$NameFromTwitterAPI <- NA
d$NameFromFacebookAPI <- NA
```

Now, let's parse out Twitter/Facebook handles from the search result links. The extracted handles will be put in the columns: *TwitterHandle* and *FBHandle*.

```
d$TwitterHandle <- tolower(d$Twitter)
d$TwitterHandle<-gsub("https://twitter.com/", "",d$Twitter)
d$TwitterHandle<-gsub("/$", "",d$TwitterHandle)
d$TwitterHandle<-gsub("…/", "",d$TwitterHandle)
d$TwitterHandle<-gsub("-", "",d$TwitterHandle)
d$TwitterHandle<-gsub(" ", "",d$TwitterHandle)
```

```
d$FBHandle <-tolower(d$Facebook)
d$FBHandle<-gsub("https://www.facebook.com/", "",d$FBHandle)
d$FBHandle<-gsub("/$", "",d$FBHandle)
d$FBHandle<-gsub("…/", "",d$FBHandle)
d$FBHandle<-gsub("-", "",d$FBHandle)
d$FBHandle<-gsub(" ", "",d$FBHandle)
```

To prevent error in API requests, we will only work on cases with a complete organization name and a Twitter handle.

```
d<- d[complete.cases(d$OrganizationName),]
d<- d[complete.cases(d$TwitterHandle),]
```

Step 2: Grab Twitter/Facebook profile information from API.

We now can start grabbing Twitter bio information. I let the API connection pause for 3 seconds after each request. Notice that in this step, we are getting the name of a Twitter account (e.g., **Alliance for Justice** for @afjustice) and store it in the column *TwitterHandle*.

```
#authorize your Twitter API connection.

setup_twitter_oauth("xxxx", "xxxxx", "xxxx", "xxxx")

for (account in d$TwitterHandle[1:length(d$TwitterHandle)]){

  print(c("finding info for:",account))

  try(d[d$TwitterHandle==account,]$NameFromTwitterAPI <-
getUser(account)$name)

  print(paste("the name is:",tw_df$name, collapse = NULL))

  #d[d$TwitterHandle==account,]$NameFromTwitterAPI <-
tw_df$name

  Sys.sleep(3)
}
```

We can also collect the profile information from the Facebook API. The name of a Facebook page is stored in the column *FBHandle*.

```
#authorize Facebook API connection.

token <- fbOAuth("xxxx", "xxx", extended_permissions =
TRUE,legacy_permissions = FALSE)

for(account in d$FBHandle[1:length(d$FBHandle)]) {
```

```
 print(c("finding info for:",account))

 try(d[d$FBHandle==account,]$NameFromFacebookAPI <- fb_df<-
getPage(page=account, token, n = 1, feed = FALSE,reactions =
FALSE)$from_name)

 Sys.sleep(3)

}
```

Step 3: Calculate the match ratio.

The match ratio is defined as **the the proportion of words in an organization's name that overlap with the name of a Twitter/Facebook account.** Here is a side-by-side comparison of the organization names with the names obtained from the Twitter API.

OrganizationName	NameFromTwitterAPI
Brady Campaign to Prevent Gun Violence	Brady Campaign
Center for Study of Responsive Law	Responsive Law
National Partnership for Women and Families	National Partnership
United States Conference of Catholic Bishops	US Catholic Bishops
National Network for Immigrant and Refugee Rights	National Network
Alliance for Retired Americans	Alliance Retirees
American Immigration Control Foundation	AIC Foundation
Americans United for Separation of Church and State	Americans United
Center for Defense Information	U.S. Dept of Defense
Consumers Union of United States, Inc.	Consumers Union
League of Women Voters	LWV of the US
National Committee to Preserve Social Security and Me	National Committee
National Organization for Women	National NOW
Union of Concern Scientists	Concerned Scientists
Acton Institute for the Study of Religion and Liberty	Acton Institute
AIDS Alliance for Children, Youth, and Families	AIDS Alliance DC
American Coalition for Fathers and Children	Fathers & FamiliesSJ

Let's first calculate the match ratio for Twitter handles and store the match ratio in the new column *MatchRatioTwitter*.

```
d$MatchRatioTwitter <- NA

for (name in
d$OrganizationName[1:length(d$OrganizationName)]){
```

```
 print(c("matching text for:",name))

 string1<-
str_replace_all(iconv(d[d$OrganizationName==name,]$Organizat
ionName),"[[:punct:]]"," ")

 string1 <- unlist(str_split(tolower(string1)," "))

 string2<-
str_replace_all(iconv(d[d$OrganizationName==name,]$NameFromT
witterAPI),"[[:punct:]]"," ")

 string2 <- unlist(str_split(tolower(string2)," "))

 d[d$OrganizationName==name,]$MatchRatioTwitter <-
length(intersect(string1,string2))/length(string1)

}
```

For Facebook:

```
d$MatchRatioFacebook <- NA

for (name in
d$OrganizationName[1:length(d$OrganizationName)]){

 print(c("matching text for:",name))

 string1<-
str_replace_all(iconv(d[d$OrganizationName==name,]$Organizat
ionName),"[[:punct:]]"," ")

 string1 <- unlist(str_split(tolower(string1)," "))

 string2<-
str_replace_all(iconv(d[d$OrganizationName==name,]$NameFromF
acebookAPI),"[[:punct:]]"," ")

 string2 <- unlist(str_split(tolower(string2)," "))

 d[d$OrganizationName==name,]$MatchRatioFacebook <-
length(intersect(string1,string2))/length(string1)

}
```

We still need human intelligence to check the remaining incomplete cases, but the process of cross-checking is made much easier with the R code.

Question? Follow and contact my Twitter @cosmopolitanvan and my website.