

A10-api-twitter

Verena Haunschmid

29 May 2016

```
library(ggplot2)
library(twitterR)
library(streamR)
```

```
## Loading required package: RCurl
```

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

```
## Loading required package: rjson
```

Twitter Streaming API

Twitter has a really great [API](#) with lots of documentation. You have to distinguish between the normal API and the [streaming API](#). From how I understood it with the normal API you can query things that have already happened (all tweets from a user, followers of a user, ...) and with the streaming API you can query things that are currently happening (e.g. observe a hashtag or your timeline). The latter is often used by apps that provide an interface for Twitter.

In this report I show how I explored the Twitter Streaming API following the hashtag “#AUTNED” that was used by Twitter users during the soccer match Austria - Nederland.

Accessing the Twitter API

Previously I have written a blog post on how to [use R to connect to twitter and create a wordcloud of your tweets](#). There I used the normal API. This time I want to use the streaming API because probably there are too many tweets written during the time of the match and the normal API might restrict it. User Ranthony from the Learning Club also created a nice [report about this](#).

```
# needed for API call
requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
```

With `Sys.setenv("consumerKey"="myConsumerKey")` and `Sys.setenv("consumerSecret"="mySecretKey")` you can set your keys as environment variables so they are not seen in the code. For this you need a Twitter App, I explained how to do this in a <http://blog.haunschmid.name/use-r-to-connect-to-twitter-and-create-a-wordcloud-of-your-tweets/>. I use environment variables because then it can't happen that I accidentally save the keys in a file and upload it to github.

It is important that you create a **new app** and don't use the same as for the normal API. At the beginning the code below didn't work for me because for other app I needed to set a callback URL, which is different for the streaming API.

```
my_oauth <- OAuthFactory$new(consumerKey = Sys.getenv("consumerKey"), consumerSecret = Sys.getenv("consumerSecret"),
  requestURL = requestURL, accessURL = accessURL, authURL = authURL)
my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
```

After running this code, a browser window should appear.

Step 1: Authorize App

Authorize veRenaStReam to use your account?

Authorize app


Cancel

This application will be able to:

- Read Tweets from your timeline.
- See who you follow.

Will not be able to:

- Follow new people.
- Update your profile.
- Post Tweets for you.
- Access your direct messages.
- See your Twitter password.




veRenaStReam
blog.haunschmid.name
used for my R applications that access the Twitter Streaming API

Step 2: PIN Code

You've granted access to veRenaStReam!

Next, return to veRenaStReam and enter this PIN to complete the authorization process:



Step 3: Enter PIN Code

```
> my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
To enable the connection, please direct your web browser to:
https://api.twitter.com/oauth/authorize?oauth_token=
When complete, record the PIN given to you and provide it here:
```

Afterwards your app should be authenticated.

Streaming the tweets

At first I wanted to observe the stream for 5 minutes and then download the tweets and save them. You can do this by setting the `time` parameter to 300 seconds. But then I noticed that the exact time is not saved for those tweets. To get a finer granularity for my analyses, I opened and closed the stream every minute.

The reason why I saved the tweets as files and not directly in a `data.frame` was simply because I didn't trust my RStudio sessions enough. And when using the streaming API you really just have one chance. I have to admit, the `while(TRUE)` is ugly and I don't usually do this, but I didn't know how long the match would be and so I simply started a never ending loop and stopped the code with `Ctrl + C` after the game ended. I started this loop about 40 minutes before the match (this is important because the time is not saved).

```
i <- 1
while (TRUE) {
  file <- paste0("10-api/tweets/tweets", i, ".json")
  track <- "AUTNED" # Austria vs. Netherlands (soccer)
  follow <- NULL
  loc <- NULL
  lang <- NULL
  minutes <- 1
  time <- 60 * minutes
  tweets <- NULL
  filterStream(file.name = file, track = track, follow = follow, locations = loc,
    language = lang, timeout = time, tweets = tweets, oauth = my_oauth,
    verbose = TRUE)
  i <- i + 1
}
```

Parse tweets

In the next step I loop over all the small files and save the tweets with an additional column `moment` to a `data.frame`.

```
n <- 157 # I checked that in the directory
tweets.df <- NULL
for (i in 1:n) {
  file <- paste0("10-api/tweets/tweets", i, ".json")
  new.tweets <- NULL
  tryCatch({
    new.tweets <- parseTweets(file, verbose = FALSE)
    new.tweets$moment <- i
  }, error = function(e) {
    message(paste(i, e))
  })
}
```

```

})

tweets.df <- rbind(tweets.df, new.tweets)
}

```

```
## 5 Error in results.list[[1]]: subscript out of bounds
```

```
## 6 Error in results.list[[1]]: subscript out of bounds
```

When are most tweets written

The first thing I wanted to check was when user tended to tweet about the game.

There are some events that might be interesting: * The game started at 20:35 (8:35 PM) which is around moment 43 or 44 * The first goal was shot in minute 9 Vincent Janssen * There is a break after about 45 minutes for about 15 minutes * There was a second goal in minute 66 by Georginio Wijnaldum * The game ends after 90 minutes + 4 minutes overtime

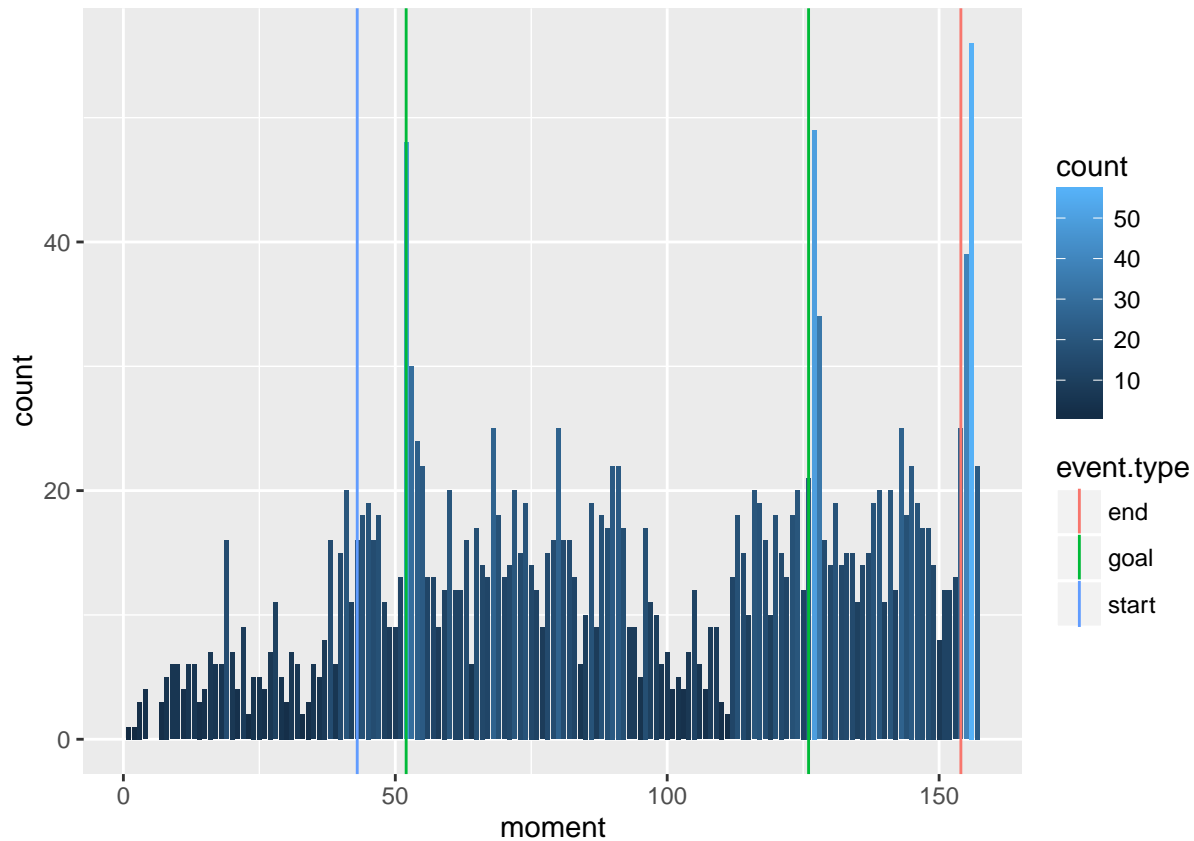
```

start_tweet <- 43
goal_1 <- 9
goal_2 <- 66
hftime_break <- 17 # guessed from the tweets

events <- data.frame(tweet.nr = c(start_tweet, start_tweet + goal_1, start_tweet +
  hftime_break + goal_2, start_tweet + hftime_break + 90 + 4), event.type = c("start",
  "goal", "goal", "end")) # ggplot is easiest to use with data.frames

ggplot(data = tweets.df, aes(x = moment)) + geom_bar(aes(fill = ..count..)) +
  geom_vline(data = events, aes(xintercept = tweet.nr, colour = event.type),
    show.legend = TRUE)

```



As I expected, most tweets are sent after the goals and after the match ends.

Possible ideas

There are many things that could be done with this dataset, e.g.:

- Look for players names - when did user talk about them (after a goal, a yellow/red card, ...)
- Which languages did users use to tweet

Useful links

I found some links that made it possible to create this document like I wanted it.

- [Working directory is forced for each chunk?](#)
- [Error line numbers are wrong](#)
- [Add vline to existing plot and have it appear in ggplot2 legend?](#)