**A short Twitter-mining and network visualization tutorial**

**Bas Hofstra & Rense Corten, October 2016**

In this short tutorial we are going to guide you through downloading Twitter-data, which is relatively easy! One can easily download Twitter-user networks and Tweets containing specific words. However, one must keep into account the shortcomings of the Twitter-rest-application-programming-interface (API). For instance, when you try to download a large number of specific *hashtags* (e.g., #pegida), the API will not actually give you *all* the tweets that contain this specific word, only a small subsample. Furthermore, you can only do a small number of requests to the API per minute before you exceed the *rate limit*, which means that you have to carefully think what kind and the size of data you want to download. Finally, the Twitter API only returns tweets from at most 9 days.

**Note:** when R or Gephi is not available at the computers in the computer lab you can easily install both programs on you USB stick and run it from there. R-Studio is definitely not available in computer lab PCs.

**Download R here:** <https://cran.r-project.org/bin/windows/base/>

**Download Gephi here:** [http://gephi.github.io/users/download/](http://gephi.github.io/users/download/%29)

**Download R-Studio here:** <https://www.rstudio.com>.

**WEEK 1: follow these steps in the *first* computer lab**

1. Make a Twitter account:

Note: if you already have a Twitter-account, you can skip this step. If you do not want to use your regular Twitter-account for this computer lab, you can make a new Twitter-account, where you’ll need to sign up with another e-mail address than the one you use for your regular Tweets. Follow the steps below:

* 1. Go to [www.twitter.com](http://www.twitter.com)
	2. Make a new Twitter-account
	3. In order to be able to download Twitter-data you must declare a telephone number. Please do so by going to settings 🡪 your Twitter data 🡪 add phone number. Wait for the verification code and add the phone number to your account. (note: sometimes there are problems with leading zeros, e.g., +31624587121 will work, but +31**0**624587121 will **not** work).
	4. Congratulations, you’re now officially a member of the Twitter community which is over 500 million strong, and more importantly; this is the first step in downloading Twitter-data
1. Make a Twitter-app:

One way to mine Twitter-data is via the statistical program R through the Twitter-rest-API. This is also the method we are going to use in this computer lab (for other methods, see e.g., Ackland p73). Therefore, you need to make a Twitter-app! This sounds more difficult than it is, we will guide you through making this app via the steps below:

* 1. Go to <https://apps.twitter.com/>
	2. Sign in with your (newly created) Twitter-account
	3. If you have never developed an application for Twitter, the new screen should say “You don’t currently have any Twitter apps”.
	4. Press “Create New App”
	5. In the new screen you have to fill out a number of fields, you can for instance name your new app “ISMN Twitter mining”, or whatever name you deem necessary. Also fill out the description as you find suitable, for instance some information about the course and why you need to make this app. The website can be *any* website that works, for instance you linkedin/Facebook/personal website; as long as the URL works (include “http://www.” in the link). Agree to the terms and conditions and create the app.
	6. Congratulations, you have created a Twitter application!
1. Configure the statistical program R

To download Twitter-data, we are going to work with the statistical package R, which can be considered a more technical version of SPSS (i.e., you can only work with syntax). Don’t worry; we are going to guide you through the process. On R’s website you can find more information about the program: <https://www.r-project.org/> . (Note: it is best to use a better version and lay-out of R, therefore use the R-studio add-on for R, which is a convenient visual representation of R, e.g., you can see which packages are loaded and which data sets are declared. R-studio also works slightly better with point and click). Download the Twitter-mining R-script from blackboard (XXXpathXXX) in a folder of your choice (e.g., designated course folder).

* 1. Open R on your computer (it is important that you do *not* run the 64-bit version; this will not work with the standard 32-bit Java that most Windows computers are equipped with). Better is to run the syntax from R-studio, which is nicer visual representation of the R-program.
	2. R is an open source program that works with *packages*, this means that R-users can build new functions and add these to the libraries in R. Users that are interested in such packages can in turn download and add these packages to R. We are definitely interested in a few packages, since they help us download Twitter-data.
	3. First we are going to need to install a few packages and load them into R. You can do this by following the “### C ###” steps in the R-script that you’ve downloaded.
	4. We now have created a situation in which we can actually start to download twitteR data through the “twitteR” package.
1. Download selections of Twitter-data

We are now going to download Twitter-data through the package twitteR. There are many more ways to get data through this package, you can always visit the help page of this package here: <https://cran.r-project.org/web/packages/twitteR/twitteR.pdf>

* 1. You must now gain access to the Twitter-API via your R-console.
	2. Go to: <https://apps.twitter.com/>
	3. Select the application that you’ve created under step 2
	4. Click the tab “Keys and access tokens”
	5. Click “Create access token”
	6. Now you can see your Consumer Key (API Key), Consumer Secret (API Secret), Access Token and Access Token Secret. These need to be copied into your R-script: please follow the “### F ###” steps in the R-script.
	7. Now that we are able to download Twitter-data, we need to first explore some functions of the twitteR package that are useful
	8. We can download mentions of specific words, which is done with the function “xxxx <- searchTwitter(“lookupword”, n = xxxx, retryOnRateLimit = xxxx)”.
	9. In this way, we can download Tweets containing specific words, but we will also know who made the original Tweet, who retweeted the Tweet, the screennames of users, sometimes the geo-location from where the Tweet is done, the platform (e.g., iPhone) from where the Tweet was done, and some additional information.
	10. For this tutorial we are interested in information surrounding the hashtag “#somethinyouwanthere”. (note: change to something you find interesting.)
	11. Go back to the R-script and follow the steps under “### K ###”.
	12. Now that we have created a dataset in excel format, open this dataset in SPSS or the statistical software of your choice (first, open the data in excel and give the first column a name and save the excel-file again, or else SPSS won’t open the data).
	13. **Question: How many Tweets have you downloaded?**
	14. **Question: How many retweets on average does a Tweet get?**
	15. **Question: How many *original* Tweets have you downloaded? (tip: select only if *isretweeted* is FALSE)**
	16. Now we want to see whether we can obtain a network from this dataset. Consider that on average Tweets are Retweeted more than five times. You might see this as directed relationships; those who Retweet an original Tweet have a tie with the original Tweeter. Also, those who reply on a Tweet have relationships to another Tweeter, we do not take these into account for this tutorial. So what we want to do now is to get a dataset that shows the directed Retweet network of the hashtag “#somethinyouwanthere”. Therefore, we need to do some cleaning on the data in R, i.e., to obtain screennames from the Tweets (we already have the original and Retweet screennames.
	17. Save your data somewhere you remember and take the data with you next week to the lab.

**WEEK 2: follow these steps in the *second* computer lab**

1. Making network data from the downloaded Twitter data last week
2. Go to the R-script of last week “ISMN\_twitterminingbh03.r” and follow the steps in “### A ###”.
3. Now we have created an actorlist and an edgelist. Open each file with a text editor and have a look. As often with social network data, the information is separated into two files:
	1. **yourfilename\_actors.txt** contains information about the actors (in this case, high school students), such that each row in the data refers to a person.
	2. **yourfilename\_ties.txt** contains information on relations between the students: each row in the data refers to a tie from one student to another. The numbers for the students are identical to those in the column “Id” in yourfilename\_actors.txt.
4. **Question: how many actors and how many ties are in the data?**
5. Loading the data into Gephi
	1. Check if Gephi is installed on your computer. If yes, open it and proceed with step 6!
	2. Download “gephi.zip” from Blackboard
	3. Unpack it somewhere, navigate to bin/gephi.exe and start the program
	4. Start new project
	5. Go to “Data laboratory”
	6. Choose “import spreadsheet”
	7. Choose yourfilename\_actors.txt
	8. Set the right separator (tab/semicolon/etc.), and set as “nodes table”
	9. Click next and then “finish” 🡪 you should now see a neat spreadsheet
	10. You’ll note that the “Label” column is still empty. We’ll fill it simply by using the values of “Id”. At the bottom of the screen, select “Copy data to other column”, then Id, and then Label.
	11. Now we need to import the tie data. Click “import spreadsheet” again.
	12. Repeat the process for the file with ties, but this time choose “edges table” as the table type. Click “next”, then “finish”.
6. Visualizing the data
	1. Go to the “Overview” tab. You should now see your network.
	2. Right now, you don’t see much structure. A visualization layout can help. In the “Layout” section on the left, choose “Fruchterman Reingold” and click run. If it doesn’t stop automatically after a while, click “stop”.
	3. You may zoom in and out by using your mouse’s scroll wheel. If necessary you can also re-center the picture by clicking the loupe icon.
	4. **Question: how many components do you see in the network?**
7. Computing some network measures
	1. On the right hand side, you’ll find the “statistics” section, which allows you to compute a range of network measures. To start, let’s compute the overall density of the network. (Think first: how is network density defined, in general?). Click the “run” button next to “Graph density”. Because this is a directed network, stick to “directed” in the next dialog.
	2. **Question: what is the density of this network?**
	3. Next, compute average shortest path length (how is that defined?) by running the “Network Diameter” procedure. Note that Gephi will, besides the average path length, also report a number of other measures. The reason is that computing paths (in large networks) takes a lot of computing time, and it is therefore efficient to compute some other measures that rely on path lengths at the same time. Also have a look at these other measures.
	4. **Question: what is the average path length of this network?**
	5. Also run “Average degree” and “Avg. clustering coefficient”.
	6. **Question: what the average degree of this network?**
	7. **Question: what the average clustering coefficient of this network?**
8. Looking at Individual-level network measures
	1. So far we’ve been looking at *global* network measures (for the entire network). However, while you computed some of those measures, Gephi automatically also computed some measures at the individual level. You can see them when you go back to the “Data laboratory” and look at the “Nodes” table.
	2. **Question: which individual-level network measures have been computed so far?**
	3. We can also use some of these measures in the visualization. Go back to “Overview”. In the upper left section, go to the “Ranking” tab and then click the “refresh” icon again. In the drop-down menu a number of variables appear on which the nodes can be ranked. Click “InDegree”, as a measure of popularity (why?). Then click the red “prism” icon, and “apply”. What happens? (Hint: you can make the differences more pronounced by increasing the number for “max size”.)
	4. On the “preview” tab, you can get really pretty versions of your layout that you can use in papers and presentations. Play around with some of the presets and settings (press the “refresh” button to apply your changes), and then export your favorite version as pdf (click “export” at the bottom of the screen).
9. Some further analyses
	1. If you use some community detection: do you see clusters of Apple and non-Apple users?C
	2. We want to test the hypothesis whether Apple users are more likely to be central in a network (i.e., their tweets are retweeted more often; having a higher indegree) than non-Apple users.
		1. Export the Gephi dataset with some centrality measures to .csv
		2. Open it in your favorite statistical software
			1. You still have to do some cleaning (i.e., not all strings in the “client” variable follow apple/nonapple; set everything that is not yet coded correctly to “nonapple” and also recode this variable to numeric values (SPSS syntax: *recode client (“apple”=1) (“nonapple”=0) (else=0) into clientx.*)
	3. **Question: Perform a hypothesis test: is there a difference in popularity for apple/nonapple users?**
	4. **Question: Perform a hypothesis test: no apple/nonapple users retweet more often?**
	5. You now have created your own personal Twitter data set, via R and have been able to produce a network and even tested a simple hypothesis!
		1. You can play around with the twitteR-package. For instance, try to download *all* the Tweets of a specific user (e.g., our prime minister or Barack Obama) or try to download all the followers/following of a specific user (e.g., specific companies: KLM), etc.
		2. A step further would be to actually try and do a sentiment analysis on the words in the Tweets. More information on how to do that, can be found here: <http://www.r-bloggers.com/twitter-sentiment-analysis-with-r/>