Introduction to netlangr

CSQ Siew

2018-01-22

This vignette demonstrates how to use the main functions in the `netlangr` package.

The `makelangnet` and `makemultinet` functions are used to create a language network from a list of words.

The `getnetstats` and `getmultistats` functions are used to compute the network measures from a language network.

The `makelangnet` and `getnetstats` functions are for single-layer networks, i.e., either phonological or orthographic similarity networks.

The `makemultinet` and `getmultistats` functions are for multi-layer networks, i.e., the phono-graphic multiplex where both phonological and orthographic links are represented in the network.

The package is still very much a work in progress, so any feedback, comments, and suggestions are very welcome!

Set up

Download the package from my github page.

```
# install.packages('devtools')
# library(devtools)
# install_github('csqsiew/netlangr')
library(netlangr)
```

Load some example data

All you really need to get started is a list of words. Spellings if you want to construct an orthographic network, and phonological transcriptions if a phonological network is desired. Note that the phonological transcriptions must be constructed such that **1 phoneme = 1 character**, this is because the networks are constructed based on edit distance of 1 (i.e., a link is placed between pairs of words that differ by the substitution, deletion, or addition of one phoneme/letter, which is the way that phonological or orthographic similarity is typically operationalized in the psycholinguistic literature; Luce & Pisoni, 1998; Coltheart et al., 1977).

```
data <- read.csv('cat.csv', stringsAsFactors = F)
# stringsasFactor = F to force the columns to character class, instead of
factor
head(data)
```

```
##   Phono Ortho
## 1   k@t   cat
## 2    @t    at
## 3   b@t   bat
## 4   k@b   cab
## 5   k@S cache
## 6   k@d   cad

class(data$Phono) # should be 'character'

## [1] "character"

class(data$Ortho) # should be 'character'

## [1] "character"
```

Single-layered network

```
# Phonological network
phono.net <- makelangnet(data$Phono) # make the language network
phono.net.measures <- getnetstats(phono.net) # get network measures
head(phono.net.measures)

##   node location degree clustering closeness_gc
## 1  k@t        G     35  0.2588235    0.8541667
## 2   @t        G     15  0.7523810    0.5694444
## 3  b@t        G     13  1.0000000    0.5540541
## 4  k@b        G     10  1.0000000    0.5189873
## 5  k@S        G     10  1.0000000    0.5189873
## 6  k@d        G     10  1.0000000    0.5189873

# Orthographic network
ortho.net <- makelangnet(data$Ortho) # make the language network
ortho.net.measures <- getnetstats(ortho.net) # get network measures
head(ortho.net.measures)

##   node location degree clustering closeness_gc
## 1  cat        G     27  0.2962963    0.9090909
## 2   at        G     12  1.0000000    0.6000000
## 3  bat        G     12  1.0000000    0.6000000
## 4  cab        G      8  1.0000000    0.5454545
## 5  cad        G      8  1.0000000    0.5454545
## 6  cam        G      8  1.0000000    0.5454545
```

Network measures:

location: G = largest connected component (giant component), L = lexical island, H = hermit

degree: number of words that are neighbors of a given word (i.e., neighborhood size)

clustering: the extent to which a word’s neighbors are also neighbors of each other, i.e., clustering in a word’s neighborhood in the network, ranges from 0 to 1.

closeness.gc: normalized inverse of the average distance between a given word and all other words in the LCC, higher values indicate that a word is close to many other words in the network (more central). (Note that closeness centrality is only calculated for words in the LCC.)

Multi-layered network

```
multi.net <- makemultinet(data)
multi.net.measures <- getmultistats(multi.net)
head(multi.net.measures)

##        node location degree.pg degree.all clustering.pg
## 1   cat;k@t        G        21         41     0.2857143
## 2    at;@t        G        12         15     0.6969697
## 3   bat;b@t        G        10         15     1.0000000
## 4   cab;k@b        G         5         13     1.0000000
## 5 cache;k@S        G         0         10     0.0000000
## 6   cad;k@d        G         5         13     1.0000000
##   clustering.unweighted clustering.weighted closeness.gc.weighted
## 1             0.2463415           0.2612903             0.6612903
## 2             0.9428571           0.9523810             0.3660714
## 3             0.9428571           0.9657143             0.3727273
## 4             0.8076923           0.8611111             0.3628319
## 5             1.0000000           1.0000000             0.4659091
## 6             0.8076923           0.8611111             0.3628319
##   closeness.gc.unweighted
## 1               1.0000000
## 2               0.6119403
## 3               0.6119403
## 4               0.5942029
## 5               0.5694444
## 6               0.5942029

# write.csv(multi.net.measures, file = 'output.csv') # export the data if you
wish
```

Network measures:

location: G = largest connected component (giant component), L = lexical island, H = hermit

degree.pg: number of words that are both phonologial AND orthographic neighbors of a given word (i.e., phonographic neighbors; the neighborhood size of the phonographic network)

degree.all: number of words that are phonological or orthographic neighbors of a given word (note that phonographic neighbors are not double counted, i.e., the neighborhood size of the phonographic multiplex)

clustering.pg: the extent to which a word's phonographic neighbors are also phonographic neighbors of each other, i.e., clustering in a word's neighborhood in the phonographic network, ranges from 0 to 1.

clustering.unweighted: the extent to which a word's phonological and orthographic neighbors are phonological or orthographic neighbors of each other, i.e., clustering in a word's neighborhood in the phonographic multiplex, ranges from 0 to 1. unweighted = each link has the same weight.

clustering.weighted: the extent to which a word's phonological and orthographic neighbors are phonological or orthographic neighbors of each other, i.e., clustering in a word's neighborhood in the phonographic multiplex, ranges from 0 to 1. weighted = phonographic links are double weighted as compared to phonological or orthogrpahic only links.

closeness.gc.unweighted: normalized inverse of the average distance between a given word and all other words in the LCC, higher values indicate that a word is close to many other words in the network (more central). unweighted = each link has the same weight.

closeness.gc.weighted: normalized inverse of the average distance between a given word and all other words in the LCC, higher values indicate that a word is close to many other words in the network (more central). weighted = phonographic links are double weighted as compared to phonological or orthogrpahic only links. (Note that closeness centrality is only calculated for words in the LCC.)

A pretty network figure

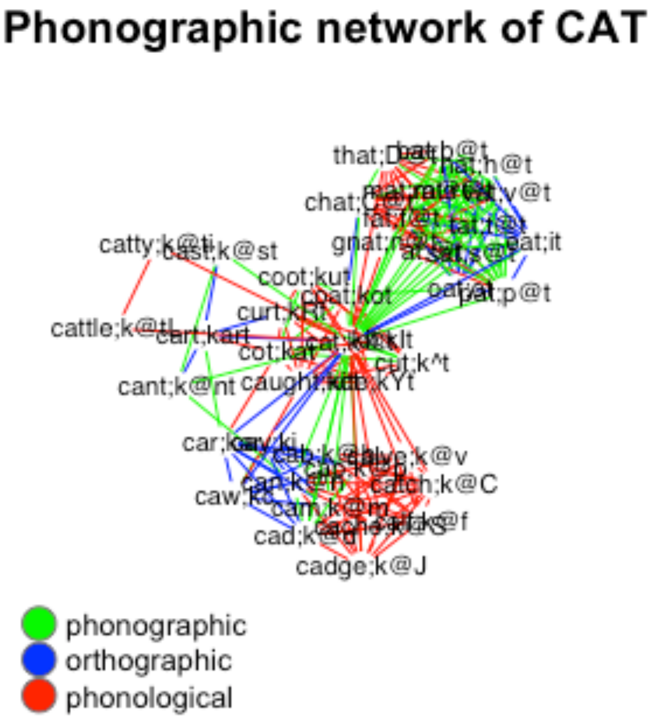
```
library(igraph)

l <- layout_with_lgl(multi.net)

# color edges by their connection type
E(multi.net)$color <- E(multi.net)$type
E(multi.net)$color <- E(multi.net)$color %>% gsub('po', 'green', .) %>%
  gsub('o', 'blue', .) %>% gsub('p', 'red', .)

plot(multi.net,
     vertex.label.color = 'black',
     vertex.color = 'white',
     vertex.label.family = 'Helvetica',
     layout = l,
     edge.color = E(multi.net)$color,
     vertex.label.cex = .7,
     vertex.shape="none", vertex.label=V(multi.net)$label,
     main = 'Phonographic network of CAT')

legend(x=-1.5, y=-1.1, c("phonographic","orthographic", "phonological"),
pch=21,
       col="#777777", pt.bg=c('green', 'blue', 'red'), pt.cex=2, cex=.8,
bty="n", ncol=1)
```


Phonographic network of CAT

References

- Phonological clustering coefficient in SWR: Chan & Vitevitch (2009)
- Closeness centrality in SWR: Goldstein & Vitevitch (2017)
- Network components in SWR: Siew & Vitevitch (2016)
- Small world networks, clustering coefficient: Watt & Strogatz (1998)
- Centrality measures: Borgatti & Everett (2006)
- Multilayer networks: Boccaletti et al., (2014), Menichetti et al., (2014)
- Neighborhood density/degree: Luce & Pisoni (1998), Coltheart et al. (1977), Andrews (1997)
- Phonological language network: Vitevitch (2008)