



R-code for network analysis and qualitative discourse analysis of a classroom group-discussion

Bruun, Jesper

Publication date:
2018

Document version
Other version

Document license:
[CC BY-ND](#)

Citation for published version (APA):
Bruun, J. (2018). R-code for network analysis and qualitative discourse analysis of a classroom group-discussion.

R-code for network analysis and qualitative discourse analysis of a classroom group-discussion

Jesper Bruun

10/19/2017

How to prepare files for loading

Initially, the files were prepared in the .doc format. Reference to students were removed. Each utterance from a student was followed by an empty line. Here is an example:

--- Begin example ---

ja vad sa vi mer vi sa nån siffra ungefär fyrahundrafemtio kanske nånting sånt, och sen var det argument för att vi inte vill ge mer fyrahundrafemtio

vad var det fyrahundrafemtio till

sexhundra nånting

vargar är jättefina djur

ja det är de absolut

... --- End example ---

We used free software to convert all .doc files to .txt files, and subsequently used a text-editor to convert all empty lines to a word that signified line change. Here we used “NewTurn”. This changed the formatting of the file:

--- Begin example new formatting --- ja vad sa vi mer vi sa naan siffra ungefaer fyrahundrafemtio kanske naanting saant, och sen var det argument foer att vi inte vill ge mer fyrahundrafemtio newTurn vad var det fyrahundrafemtio till newTurn sexhundra naanting newTurn vargar aer jaettefina djur newTurn ja det aer de absolut NewTurn ... --- End example new formatting ---

Here, we would now implement changes to words. This can be done via R and the “gsub” command, for example “allFilesx<-gsub(” anvaends ”, ” anvaend ” ,allFilesx, fixed=T)“. However, in our experience, this command does not work well with the tm-package. Therefore, we recommend that these substitutions be done before using the tm-package – for example, by using a script in a text-editor.

This is what we have done in our example text for this document.

How to load a set of transcription files into R for text-mining

Here is the code we used to load a corpus of files. A corpus could be a set of .txt documents, each of which is a transcript of the sort shown above.

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(igraph)
```

```
##  
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':  
##  
## decompose, spectrum
```

```
## The following object is masked from 'package:base':  
##  
## union
```

```
#Set working directory to where the your folder with data is.  
#Next load all transcript-files using the Corpus command from the tm-package  
allFilesx <-Corpus(DirSource("example", encoding="UTF-8"), readerControl = list(language="lat"))  
#Next use standard text-mining functions:  
allFilesx<-tm_map(allFilesx,tolower) #Changing lower case  
allFilesx<-tm_map(allFilesx,removePunctuation) #emoving commas, punctums, and so forth  
allFilesx<-tm_map(allFilesx,stripWhitespace) # Remove excess white spaces  
allFilesx #An overview of all the loaded transcripts
```

```
## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 1
```

```
allFilesx[[1]][1] #Shows a particular transcript - in this case the only one
```

```
## $content
## [1] "<U+FEFF>jag kan tycka saa haer att eftersom att vargen finns naturligt
i sverige saa ska den vael aendaa faa finnas kvar newturn raetten till att goe
ra det men mer eller mindre newturn jag vet inte det spelar vael inte saa stor
newturn det aer vael mer kontrollera foer det aer ju faktiskt saa att de som a
er beroerda av det vill ju inte kan inte leva ett normalt liv de vill ju inte
slaepa ut sina barn foer att gaa till bussen eller saa utan alltsaa det gaell
er att faa det kontrollerat eller vad man ska saega newturn det aer ju svaart
att veta var varje varg haaller till newturn men jag tror aendaa att trehundra
som det aer idag verkar ju inte vara naagra jaettestora problem men hade man h
aft det hade man ju hoert mer newturn hade det varit fler hade man newturn typ
sjuhundra som det staar daer newturn det tror jag hade vart mycket newturn och
daa tror vi att de kommer newturn tror att de kommer laengre ner hit i sverige
ocksaa newturn tror ju det som det var nu ungefaer laater ju ganska bra det so
m han sa trehundra naanting newturn men det kaenns som man kan kontrollera det
mer newturn det aer ju naestan hundratjugo som finns idag newturn det aer vael
lite synd inavlade newturn jag fattar inte newturn men daa faar man vael likso
m haemta fler vargar newturn det staar ju det att daa minskas vargarna newturn
jo fast det aer ju ryssland aer saa otrolig stort de har jaettemycket vargar d
aer saa det aer inga problem saa aeven om de tar kanske trettio vargar saa kom
mer det ju inte att vara naagra problem foer det aer saa sjukt mycket daer new
turn det gaar ju eftersom de tagit saa haer ett omraade raett saa stort omraad
e daer de har varg och kommer newturn ja naat saant skulle ju vara foer liksom
newturn de har de vael saant daer newturn som han sa igaar med det haer med re
nskoetare att naan slags foer det aer ju det som aer stora problemet det aer j
u samerna som verkligen vill bli av med resten av sveriges befolkning newturn
men om man har saa ett vargomraade om man faar reda paa det daa kan ju samerna
haalla borta sina renar daerifraan newturn men man kan ju istaellet foer att s
atsa paa att minska vargen kan vi satsa paa att skydda renarna newturn ja men
att foersoeka faa kontrollera vargen lite mer kanske newturn det borde ju aend
aa vara laettast newturn det skulle ju vara mest snaellt mot naturen paa det v
iset newturn men jag menar om de tar naat omraade som liksom har stor skog och
inom kort fraan en stad eller naat och samerna faar reda paa det daa haaller d
e ju borta sina renar daa aer ju problemet loest newturn eller i alla fall det
problemet newturn ja saa gaeller det ju daa att de inte vandrar daa vargarna n
ewturn foer de soeker sig ju till mat newturn alla tycker ju olika i den haer
newturn saa det aer inte saa enkelt att svara newturn speciellt naer vi inte a
er saa newturn beroerda av det egentligen newturn nae och vi bryr oss ju inte
saa mycket newturn vi aer ju haer nere liksom newturn hade vi sett naan newtur
n fast om det blir foer maanga kommer de ju ner hit till oss ocksaa saa newtur
n ja de faar inte bli foer maanga foer daa finns det inte tillraeckligt med ma
t newturn jag tror inte det finns risk foer att det blir foer maanga heller ne
wturn eftersom inavlade inte lever lika laenge daa sen att newturn det tar ju
ett tag ocksaa foer att bli saa pass maanga endoffile"
```

How to text-mine those files

Next, we defined the words we wanted to remove. We call them joining words, because in later iterations, we want to keep these words as properties of the links.

```

joiningWords<-c("i","det","aer","ju","att","inte","saa","de","jag","man","och"
,"men","ja","foer","som","om","kan","daa","vi","paa","ska","har","alltsaa","va
d","en","haer","eller","med","liksom","du","vael","nae","ha","bara","ocksaa","
vara","vet","typ","skulle","vill","var","finns","ett","mycket","lite","mer","s
en","till","faar","blir","hur","maaste","ta","den","goera","alla","bra","daer"
,"kommer","goer","av","ni","saant","dem","bli","fast","ut","aendaa","hon","nae
r","tar","maanga","p","sig","andra","nu","faa","egentligen","hade","fan","saeg
er","gaar","jo","sa","saega","aendaa","hon","naer","bort","tar","maanga","p","
sig","andra","nu","faa","egentligen","hade","fan","saeger","gaar","oever","all
t","va","komma","oss","saana","ville","jaevla","skriva","kunna","aat","naagont
ing","nej","slag","naagon","aha","alltsaa","bla","ens","er","era","hej","kagg"
,"oh","om","antecknar","osv","stagg","stavas","tex","aeh","aah","aa","g","y","
m","detatt","gud","daniel","goeran","persson","herre","herregud","heter","jame
n","jaekla","jaevligt","kalmar","markus","naehae","naemen","skrev","skriv","sk
river","skrivit","tycken","vete","newturn")

```

How to convert files into networks that can be analysed

Having done this pre-processing work, we now turn to making networks. First we make an edgelist.

```

allFilesx<-tm_map(allFilesx,stripWhitespace) #You may need to clean up the cor
pus again
myWordNetwork <- function (txt,j.words) {
  #This function was created by Jonas Forsman and later adapted to this study
  # Create edge list for export
  edge.list <-strsplit(as.vector(txt)$content, split = " ")[[1]]
  # index of "declarative words" (the -1 is because we're removing first and
  # last word to make it into a word-network
  node.num <- length(edge.list[!(edge.list %in% j.words)]) - 1
  #Create matrix to be exported
  export.edge.list <- array("",dim=c(node.num,2))
  # Put source word except the "joining words into the matrix to be exported
  export.edge.list[,1] <- edge.list[!(edge.list %in% j.words)][-length(edge.li
st[!(edge.list %in% j.words)])]
  # put the target word except the "joining words into the matrix to be export
ed
  export.edge.list[,2] <- edge.list[!(edge.list %in% j.words)][-1]
  #Go through the joining words and put them in
  # for(i in j.words) {
  #   for (j in which(edge.list %in% i)) {
  #     export.edge.list[j - sum(!which(edge.list %in% j.words) > j), 3] <- i
  #   }
  #} We need a way to put multiple joining words here.
  return(export.edge.list)
}

#Now make the list of edges:
example<-as.matrix(myWordNetwork(allFilesx[[1]][1],joiningWords))
example

```

```

##      [,1]          [,2]
## [1,] "<U+FEFF>jag"  "tycka"
## [2,] "tycka"       "eftersom"
## [3,] "eftersom"    "vargen"
## [4,] "vargen"      "naturligt"
## [5,] "naturligt"   "sverige"
## [6,] "sverige"     "finnas"
## [7,] "finnas"      "kvar"
## [8,] "kvar"        "raetten"
## [9,] "raetten"     "mindre"
## [10,] "mindre"     "spelar"
## [11,] "spelar"     "stor"
## [12,] "stor"       "kontrollera"
## [13,] "kontrollera" "faktiskt"
## [14,] "faktiskt"   "beroerda"
## [15,] "beroerda"   "leva"
## [16,] "leva"       "normalt"
## [17,] "normalt"    "liv"
## [18,] "liv"        "slaepa"
## [19,] "slaepa"     "sina"
## [20,] "sina"       "barn"
## [21,] "barn"       "gaa"
## [22,] "gaa"        "bussen"
## [23,] "bussen"     "utan"
## [24,] "utan"       "gaeller"
## [25,] "gaeller"    "kontrollerat"
## [26,] "kontrollerat" "svaart"
## [27,] "svaart"     "veta"
## [28,] "veta"       "varje"
## [29,] "varje"     "varg"
## [30,] "varg"       "haaller"
## [31,] "haaller"    "tror"
## [32,] "tror"       "trehundra"
## [33,] "trehundra"  "idag"
## [34,] "idag"       "verkar"
## [35,] "verkar"     "naagra"
## [36,] "naagra"     "jaestestora"
## [37,] "jaestestora" "problem"
## [38,] "problem"    "haft"
## [39,] "haft"       "hoert"
## [40,] "hoert"      "varit"
## [41,] "varit"     "fler"
## [42,] "fler"      "sjuhundra"
## [43,] "sjuhundra"  "staar"
## [44,] "staar"     "tror"
## [45,] "tror"      "vart"
## [46,] "vart"      "tror"
## [47,] "tror"      "tror"
## [48,] "tror"      "laengre"
## [49,] "laengre"   "ner"
## [50,] "ner"       "hit"
## [51,] "hit"       "sverige"

```

```

## [52,] "sverige"      "tror"
## [53,] "tror"         "ungefaer"
## [54,] "ungefaer"    "laater"
## [55,] "laater"      "ganska"
## [56,] "ganska"     "han"
## [57,] "han"         "trehundra"
## [58,] "trehundra"   "naanting"
## [59,] "naanting"    "kaenns"
## [60,] "kaenns"     "kontrollera"
## [61,] "kontrollera" "naestan"
## [62,] "naestan"    "hundratjugo"
## [63,] "hundratjugo" "idag"
## [64,] "idag"       "synd"
## [65,] "synd"       "inavlade"
## [66,] "inavlade"   "fattar"
## [67,] "fattar"     "haemta"
## [68,] "haemta"     "fler"
## [69,] "fler"       "vargar"
## [70,] "vargar"     "staar"
## [71,] "staar"      "minskas"
## [72,] "minskas"   "vargarna"
## [73,] "vargarna"  "ryssland"
## [74,] "ryssland"  "otrolig"
## [75,] "otrolig"   "stort"
## [76,] "stort"     "jaettemycket"
## [77,] "jaettemycket" "vargar"
## [78,] "vargar"    "inga"
## [79,] "inga"      "problem"
## [80,] "problem"   "aeven"
## [81,] "aeven"     "kanske"
## [82,] "kanske"    "trettio"
## [83,] "trettio"   "vargar"
## [84,] "vargar"    "naagra"
## [85,] "naagra"    "problem"
## [86,] "problem"   "sjukt"
## [87,] "sjukt"     "eftersom"
## [88,] "eftersom"  "tagit"
## [89,] "tagit"     "omraade"
## [90,] "omraade"   "raett"
## [91,] "raett"     "stort"
## [92,] "stort"     "omraade"
## [93,] "omraade"   "varg"
## [94,] "varg"      "naat"
## [95,] "naat"     "han"
## [96,] "han"       "igaar"
## [97,] "igaar"     "renskoetare"
## [98,] "renskoetare" "naan"
## [99,] "naan"      "slags"
## [100,] "slags"    "stora"
## [101,] "stora"    "problemet"
## [102,] "problemet" "samerna"
## [103,] "samerna"  "verkligen"
## [104,] "verkligen" "resten"

```

```

## [105,] "resten"      "sveriges"
## [106,] "sveriges"   "befolkning"
## [107,] "befolkning" "vargomraade"
## [108,] "vargomraade" "reda"
## [109,] "reda"       "samerna"
## [110,] "samerna"   "haalla"
## [111,] "haalla"    "borta"
## [112,] "borta"     "sina"
## [113,] "sina"      "renar"
## [114,] "renar"     "daerifraan"
## [115,] "daerifraan" "istaellet"
## [116,] "istaellet" "satsa"
## [117,] "satsa"     "minska"
## [118,] "minska"    "vargen"
## [119,] "vargen"    "satsa"
## [120,] "satsa"     "skydda"
## [121,] "skydda"    "renarna"
## [122,] "renarna"   "foersoeka"
## [123,] "foersoeka" "kontrollera"
## [124,] "kontrollera" "vargen"
## [125,] "vargen"    "kanske"
## [126,] "kanske"    "borde"
## [127,] "borde"     "laettast"
## [128,] "laettast"  "mest"
## [129,] "mest"      "snaellt"
## [130,] "snaellt"   "mot"
## [131,] "mot"       "naturen"
## [132,] "naturen"   "viset"
## [133,] "viset"     "menar"
## [134,] "menar"     "naat"
## [135,] "naat"      "omraade"
## [136,] "omraade"   "stor"
## [137,] "stor"      "skog"
## [138,] "skog"      "inom"
## [139,] "inom"      "kort"
## [140,] "kort"      "fraan"
## [141,] "fraan"     "stad"
## [142,] "stad"      "naat"
## [143,] "naat"      "samerna"
## [144,] "samerna"   "reda"
## [145,] "reda"      "haaller"
## [146,] "haaller"   "borta"
## [147,] "borta"     "sina"
## [148,] "sina"      "renar"
## [149,] "renar"     "problemet"
## [150,] "problemet" "loest"
## [151,] "loest"     "fall"
## [152,] "fall"      "problemet"
## [153,] "problemet" "gaeller"
## [154,] "gaeller"   "vandrar"
## [155,] "vandrar"   "vargarna"
## [156,] "vargarna" "soeker"
## [157,] "soeker"    "mat"

```



```

## [158,] "mat"          "tycker"
## [159,] "tycker"     "olika"
## [160,] "olika"      "enkelt"
## [161,] "enkelt"     "svara"
## [162,] "svara"      "speciellt"
## [163,] "speciellt"  "beroerda"
## [164,] "beroerda"   "bryr"
## [165,] "bryr"       "nere"
## [166,] "nere"        "sett"
## [167,] "sett"        "naan"
## [168,] "naan"        "ner"
## [169,] "ner"         "hit"
## [170,] "hit"         "tillraeckligt"
## [171,] "tillraeckligt" "mat"
## [172,] "mat"         "tror"
## [173,] "tror"        "risk"
## [174,] "risk"        "heller"
## [175,] "heller"     "eftersom"
## [176,] "eftersom"   "inavlade"
## [177,] "inavlade"   "lever"
## [178,] "lever"      "lika"
## [179,] "lika"       "laenge"
## [180,] "laenge"     "tag"
## [181,] "tag"        "pass"
## [182,] "pass"       "endoffile"

```

The edge list here has a couple of unwanted nodes. For example, “jag”, should be “jag”. Also, “endoffile” should not be a node. So we fix these errors manually:

```

example[1,1]<-"jag"
example<-example[-182,]
example

```

```

##      [,1]      [,2]
## [1,] "jag"    "tycka"
## [2,] "tycka"  "eftersom"
## [3,] "eftersom" "vargen"
## [4,] "vargen" "naturligt"
## [5,] "naturligt" "sverige"
## [6,] "sverige" "finnas"
## [7,] "finnas"  "kvar"
## [8,] "kvar"    "raetten"
## [9,] "raetten" "mindre"
## [10,] "mindre" "spelar"
## [11,] "spelar" "stor"
## [12,] "stor"    "kontrollera"
## [13,] "kontrollera" "faktiskt"
## [14,] "faktiskt" "beroerda"
## [15,] "beroerda" "leva"
## [16,] "leva"    "normalt"
## [17,] "normalt" "liv"
## [18,] "liv"    "slaepa"
## [19,] "slaepa" "sina"
## [20,] "sina"   "barn"
## [21,] "barn"   "gaa"
## [22,] "gaa"    "bussen"
## [23,] "bussen" "utan"
## [24,] "utan"   "gaeller"
## [25,] "gaeller" "kontrollerat"
## [26,] "kontrollerat" "svaart"
## [27,] "svaart" "veta"
## [28,] "veta"   "varje"
## [29,] "varje"  "varg"
## [30,] "varg"   "haaller"
## [31,] "haaller" "tror"
## [32,] "tror"   "trehundra"
## [33,] "trehundra" "idag"
## [34,] "idag"   "verkar"
## [35,] "verkar" "naagra"
## [36,] "naagra" "jaettestora"
## [37,] "jaettestora" "problem"
## [38,] "problem" "haft"
## [39,] "haft"   "hoert"
## [40,] "hoert"  "varit"
## [41,] "varit"  "fler"
## [42,] "fler"   "sjuhundra"
## [43,] "sjuhundra" "staar"
## [44,] "staar"  "tror"
## [45,] "tror"   "vart"
## [46,] "vart"   "tror"
## [47,] "tror"   "tror"
## [48,] "tror"   "laengre"
## [49,] "laengre" "ner"
## [50,] "ner"    "hit"
## [51,] "hit"    "sverige"

```

```

## [52,] "sverige"      "tror"
## [53,] "tror"         "ungefaer"
## [54,] "ungefaer"    "laater"
## [55,] "laater"      "ganska"
## [56,] "ganska"     "han"
## [57,] "han"         "trehundra"
## [58,] "trehundra"  "naanting"
## [59,] "naanting"   "kaenns"
## [60,] "kaenns"     "kontrollera"
## [61,] "kontrollera" "naestan"
## [62,] "naestan"    "hundratjugo"
## [63,] "hundratjugo" "idag"
## [64,] "idag"       "synd"
## [65,] "synd"       "inavlade"
## [66,] "inavlade"   "fattar"
## [67,] "fattar"     "haemta"
## [68,] "haemta"    "fler"
## [69,] "fler"      "vargar"
## [70,] "vargar"    "staar"
## [71,] "staar"     "minskas"
## [72,] "minskas"   "vargarna"
## [73,] "vargarna"  "ryssland"
## [74,] "ryssland"  "otrolig"
## [75,] "otrolig"   "stort"
## [76,] "stort"     "jaettemycket"
## [77,] "jaettemycket" "vargar"
## [78,] "vargar"    "inga"
## [79,] "inga"      "problem"
## [80,] "problem"   "aeven"
## [81,] "aeven"     "kanske"
## [82,] "kanske"    "trettio"
## [83,] "trettio"   "vargar"
## [84,] "vargar"    "naagra"
## [85,] "naagra"    "problem"
## [86,] "problem"   "sjukt"
## [87,] "sjukt"     "eftersom"
## [88,] "eftersom"  "tagit"
## [89,] "tagit"     "omraade"
## [90,] "omraade"   "raett"
## [91,] "raett"     "stort"
## [92,] "stort"     "omraade"
## [93,] "omraade"   "varg"
## [94,] "varg"      "naat"
## [95,] "naat"     "han"
## [96,] "han"       "igaar"
## [97,] "igaar"     "renskoetare"
## [98,] "renskoetare" "naan"
## [99,] "naan"      "slags"
## [100,] "slags"    "stora"
## [101,] "stora"   "problemet"
## [102,] "problemet" "samerna"
## [103,] "samerna"  "verkligen"
## [104,] "verkligen" "resten"

```

```

## [105,] "resten"      "sveriges"
## [106,] "sveriges"   "befolkning"
## [107,] "befolkning" "vargomraade"
## [108,] "vargomraade" "reda"
## [109,] "reda"       "samerna"
## [110,] "samerna"    "haalla"
## [111,] "haalla"     "borta"
## [112,] "borta"      "sina"
## [113,] "sina"       "renar"
## [114,] "renar"      "daerifraan"
## [115,] "daerifraan" "istaellet"
## [116,] "istaellet"  "satsa"
## [117,] "satsa"      "minska"
## [118,] "minska"     "vargen"
## [119,] "vargen"     "satsa"
## [120,] "satsa"      "skydda"
## [121,] "skydda"     "renarna"
## [122,] "renarna"    "foersoeka"
## [123,] "foersoeka"  "kontrollera"
## [124,] "kontrollera" "vargen"
## [125,] "vargen"     "kanske"
## [126,] "kanske"     "borde"
## [127,] "borde"      "laettast"
## [128,] "laettast"   "mest"
## [129,] "mest"       "snaellt"
## [130,] "snaellt"    "mot"
## [131,] "mot"        "naturen"
## [132,] "naturen"    "viset"
## [133,] "viset"      "menar"
## [134,] "menar"      "naat"
## [135,] "naat"       "omraade"
## [136,] "omraade"    "stor"
## [137,] "stor"       "skog"
## [138,] "skog"       "inom"
## [139,] "inom"       "kort"
## [140,] "kort"       "fraan"
## [141,] "fraan"      "stad"
## [142,] "stad"       "naat"
## [143,] "naat"       "samerna"
## [144,] "samerna"    "reda"
## [145,] "reda"       "haaller"
## [146,] "haaller"    "borta"
## [147,] "borta"      "sina"
## [148,] "sina"       "renar"
## [149,] "renar"      "problemet"
## [150,] "problemet"  "loest"
## [151,] "loest"      "fall"
## [152,] "fall"       "problemet"
## [153,] "problemet"  "gaeller"
## [154,] "gaeller"    "vandrar"
## [155,] "vandrar"    "vargarna"
## [156,] "vargarna"   "soeker"
## [157,] "soeker"     "mat"

```

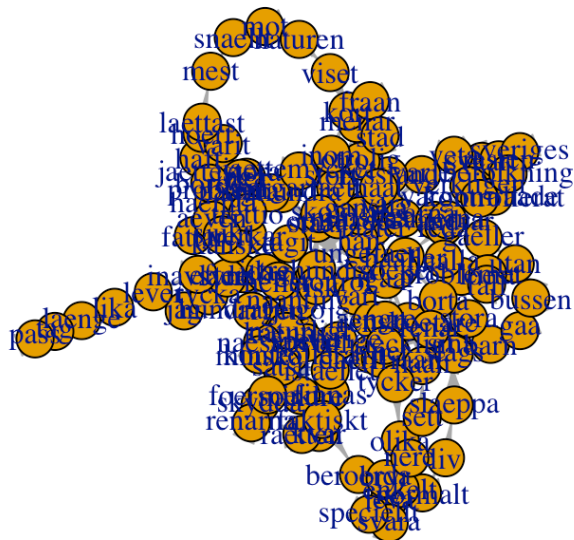
```
## [158,] "mat"          "tycker"
## [159,] "tycker"     "olika"
## [160,] "olika"      "enkelt"
## [161,] "enkelt"     "svara"
## [162,] "svara"      "speciellt"
## [163,] "speciellt"  "beroerda"
## [164,] "beroerda"   "bryr"
## [165,] "bryr"       "nere"
## [166,] "nere"       "sett"
## [167,] "sett"       "naan"
## [168,] "naan"       "ner"
## [169,] "ner"        "hit"
## [170,] "hit"        "tillraeckligt"
## [171,] "tillraeckligt" "mat"
## [172,] "mat"        "tror"
## [173,] "tror"       "risk"
## [174,] "risk"       "heller"
## [175,] "heller"     "eftersom"
## [176,] "eftersom"   "inavlade"
## [177,] "inavlade"   "lever"
## [178,] "lever"      "lika"
## [179,] "lika"       "laenge"
## [180,] "laenge"     "tag"
## [181,] "tag"        "pass"
```

#Then make a network:

```
example_net<-graph.edgelist(example,directed=T) #Creates a network from an edge list
V(example_net)$id<-V(example_net)$name
E(example_net)$weight<-1
example_net<-simplify(example_net,remove.multiple=T,edge.attr.comb="sum")
example_net
```

```
## IGRAPH 4918002 DNW- 133 177 --
## + attr: name (v/c), id (v/c), weight (e/n)
## + edges from 4918002 (vertex names):
## [1] jag      ->tycka    tycka      ->eftersom
## [3] eftersom ->vargen   eftersom   ->inavlade
## [5] eftersom ->tagit    vargen     ->naturligt
## [7] vargen   ->kanske   vargen     ->satsa
## [9] naturligt ->sverige  sverige    ->finnas
## [11] sverige  ->tror     finnas     ->kvar
## [13] kvar     ->raetten  raetten    ->mindre
## [15] mindre  ->spelar  spelar     ->stor
## + ... omitted several edges
```

```
plot(example_net)
```



The network can be exported via the `write.graph` command: `write.graph(example_net, "example.net", format="pajek)`. Subsequently, it can be used at mapequation.org

How to calculate each of the measures used in the main article

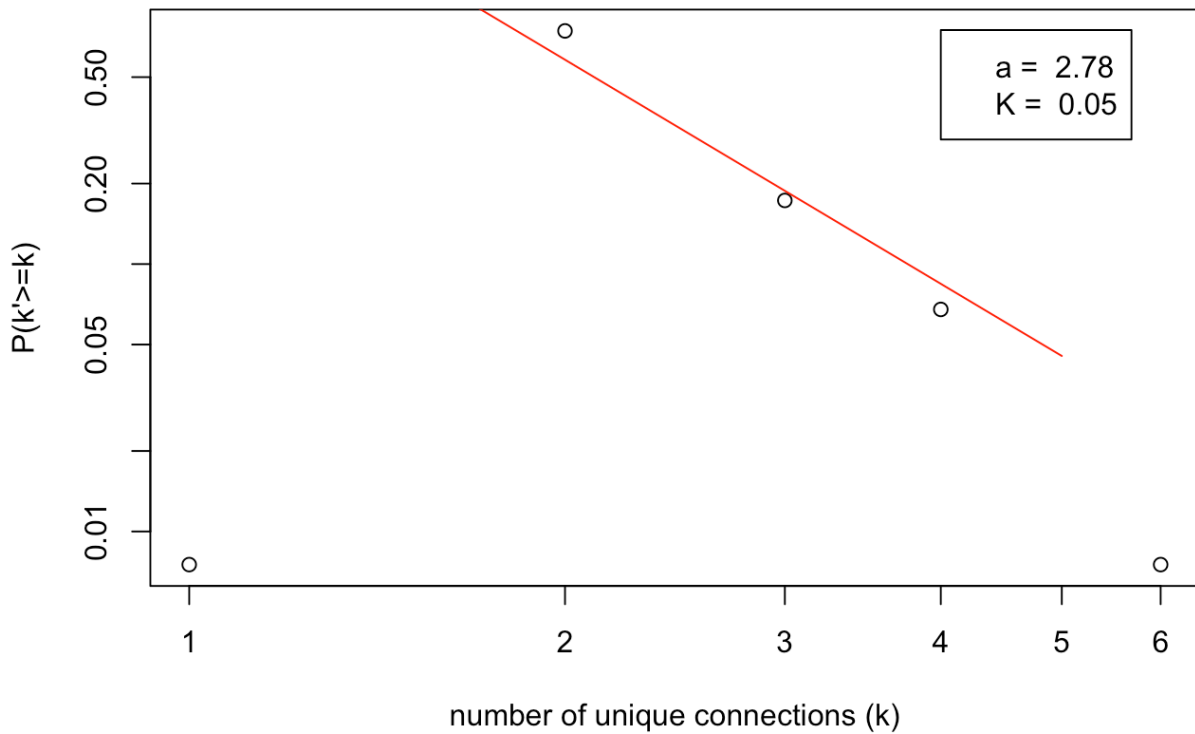
Degree distribution

```
#Degree distribution
fitt<-power.law.fit(degree(example_net,mode="in"),implementation = "plfit")
d<-degree(example_net,mode="in")
funk<-(c(1:max(d)))^(-fitt$alpha)
plot(degree.distribution(example_net,cumulative = F,mode="in"),log="xy",xlab =
"number of unique connections (k)",ylab="P(k')>=k",main="Example degree distri
bution")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 y value <= 0 omitted
## from logarithmic plot
```

```
points(4*funk,type="l",col="red")
legend(4,0.75,legend = c(paste("a = ",round(fitt$alpha,digits=2)),paste("K = "
,round(fitt$KS.stat,digits=2))))
```

Example degree distribution



```
#The measures density and mean degree (also the standard deviation can be calculated)
graph.density(example_net)
```

```
## [1] 0.01008202
```

```
mean(degree(example_net))
```

```
## [1] 2.661654
```

```
sd(degree(example_net))
```

```
## [1] 1.347766
```

```
#Diameter
diameter(example_net)
```

```
## [1] 26
```

```
#Mutual connections
length(which(which_mutual(example_net)))
```

```
## [1] 4
```

For scripts on target entropy and search information refer to: Bruun, J. & Brewé, E. (2013): Talking and learning physics: Predicting future grades from network measures and FCI pre-test scores The scripts are in the Supplemental Materials section: <https://journals.aps.org/prper/abstract/10.1103/PhysRevSTPER.9.020109> (<https://journals.aps.org/prper/abstract/10.1103/PhysRevSTPER.9.020109>)

Having loaded the scripts (use source("path-to-scripts"), for example), you use the by entering the following:

```
te_ex<-TargetEntropy(example_net)
si_ex<-sInfMatrix(example_net)

mean(te_ex)
```

```
## [1] 0.2235941
```

```
te_ex
```

```
## [1] 0.000000 0.000000 0.9122167 1.4457347 0.000000 0.9988816 0.000000
## [8] 0.000000 0.000000 0.000000 0.000000 0.8381689 1.4190573 0.000000
## [15] 0.5572222 0.000000 0.000000 0.000000 0.000000 0.7610292 0.000000
## [22] 0.000000 0.000000 0.000000 0.2018399 0.000000 0.000000 0.000000
## [29] 0.000000 0.8493728 0.9507325 1.8632421 0.6796175 0.5714500 0.000000
## [36] 0.9718643 0.000000 0.8050872 0.000000 0.000000 0.000000 0.6767354
## [43] 0.000000 0.3443002 0.000000 0.000000 0.8017832 0.000000 0.000000
## [50] 0.000000 0.000000 0.9955230 0.000000 0.000000 0.000000 0.000000
## [57] 0.000000 0.9928124 0.000000 0.000000 1.3495321 0.000000 0.9620535
## [64] 0.000000 0.000000 0.9995974 0.000000 0.000000 0.000000 0.6282474
## [71] 0.000000 0.000000 0.000000 1.3733278 0.000000 0.9432163 0.000000
## [78] 0.000000 0.7312769 0.000000 0.000000 1.0320379 1.1208562 0.000000
## [85] 0.000000 0.000000 0.000000 0.000000 0.2393965 0.000000 0.9837940
## [92] 0.000000 0.000000 0.000000 0.7389163 0.000000 0.000000 0.000000
## [99] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## [106] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## [113] 0.000000 0.000000 0.000000 0.000000 0.9990942 0.000000 0.000000
## [120] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## [127] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
```

```
mean(as.matrix(si_ex),na.rm=T)
```

```
## [1] 5.643403
```

```
si_ex[1:10,1:10] #The first ten rows and columns in the matrix
```



```
##      V1 V2      V3      V4      V5      V6      V7      V8      V9
## 1    0  0 0.000000 1.584963 3.169925 3.169925 4.169925 4.169925 4.169925
## 2  NA  0 0.000000 1.584963 3.169925 3.169925 4.169925 4.169925 4.169925
## 3  NA NA 0.000000 1.584963 3.169925 3.169925 4.169925 4.169925 4.169925
## 4  NA NA 4.906891 0.000000 1.584963 1.584963 2.584963 2.584963 2.584963
## 5  NA NA 3.321928 4.906891 0.000000 0.000000 1.000000 1.000000 1.000000
## 6  NA NA 3.321928 4.906891 6.491853 0.000000 1.000000 1.000000 1.000000
## 7  NA NA 7.491853 2.584963 4.169925 4.169925 0.000000 0.000000 0.000000
## 8  NA NA 7.491853 2.584963 4.169925 4.169925 5.169925 0.000000 0.000000
## 9  NA NA 7.491853 2.584963 4.169925 4.169925 5.169925 5.169925 0.000000
## 10 NA NA 7.491853 2.584963 4.169925 4.169925 5.169925 5.169925 5.169925
##          V10
## 1    4.169925
## 2    4.169925
## 3    4.169925
## 4    2.584963
## 5    1.000000
## 6    1.000000
## 7    0.000000
## 8    0.000000
## 9    0.000000
## 10   0.000000
```

Community detection can also be done in R, although we have used mapequation.org and the accompanying C++ code. Here, we show how to calculate communities with `infomap` in R, and how to calculate PageRanks.

```
imex<-infomap.community(example_net)
imex
```

```
## IGRAPH clustering infomap, groups: 21, mod: 0.66
## + groups:
##  $\`1`
##  [1] "naturligt"      "sverige"      "tror"         "vart"
##  [5] "laengre"        "ner"          "hit"          "mat"
##  [9] "tillraeckligt"
##
##  $\`2`
##  [1] "naagra"         "jaettestora"  "problem"      "vargar"
##  [5] "jaettemycket"  "inga"         "aeven"        "kanske"
##  [9] "trettio"
##
##  + ... omitted several groups/vertices
```

```
#Getting codelength, L
imex$codelength
```

```
## [1] 5.448328
```

```
#Calculate PageRanks  
PR<-page.rank(example_net)  
PR
```

```

## $vector
##      jag      tycka      eftersom      vargen      naturligt
## 0.001168865 0.002162401 0.012282398 0.013736405 0.005060847
##      sverige      finnas      kvar      raetten      mindre
## 0.009808588 0.005337515 0.005705753 0.006018756 0.006284808
##      spelar      stor      kontrollera      faktiskt      beroerda
## 0.006510952 0.010875593 0.015817502 0.005650491 0.011696364
##      leva      normalt      liv      slaepa      sina
## 0.006139820 0.006387712 0.006598421 0.006777523 0.015757433
##      barn      gaa      bussen      utan      gaeller
## 0.005633471 0.005957316 0.006232584 0.006466562 0.011236323
## kontrollerat      svaart      veta      varje      varg
## 0.005944302 0.006221522 0.006457159 0.006657451 0.011000117
##      haaller      tror      trehundra      idag      verkar
## 0.010592791 0.023244057 0.009843808 0.010428499 0.005600977
##      naagra      jaettestora      problem      haft      hoert
## 0.010259177 0.005529015 0.014902273 0.005391176 0.005751365
##      varit      fler      sjuhundra      staar      vart
## 0.006057525 0.011045171 0.005863063 0.010481950 0.005120355
##      laengre      ner      hit      ungefaer      laater
## 0.005120355 0.010633177 0.010207066 0.005120355 0.005521167
##      ganska      han      naanting      kaenns      naestan
## 0.005861857 0.011114006 0.005352484 0.005718476 0.005650491
##      hundratjugo      synd      inavlade      fattar      haemta
## 0.005971783 0.005600977 0.009409709 0.005167992 0.005561658
##      vargar      minskas      vargarna      ryssland      otrolig
## 0.015280520 0.005623694 0.011001662 0.005844572 0.006136751
##      stort      jaettemycket      inga      aeven      kanske
## 0.010925195 0.005812073 0.005498346 0.005391176 0.009643346
##      trettio      sjukt      tagit      omraade      raett
## 0.005267287 0.005391176 0.004648878 0.014726182 0.005341284
##      naat      igaar      renskoetare      naan      slags
## 0.017514925 0.005892318 0.006177336 0.012028258 0.006280875
##      stora      problemet      samerna      verkligen      resten
## 0.006507609 0.016132517 0.015451184 0.005546701 0.005883561
##      sveriges      befolkning      vargomraade      reda      haalla
## 0.006169892 0.006413274 0.006620148 0.011173827 0.005546701
##      borta      renar      daerifraan      istaellet      satsa
## 0.010385497 0.010098077 0.005460548 0.005810331 0.009999628
##      minska      skydda      renarna      foersoeka      borde
## 0.005418707 0.005418707 0.005774767 0.006077417 0.005267287
##      laettast      mest      snaellt      mot      naturen
## 0.005646060 0.005968016 0.006241679 0.006474293 0.006672014
##      viset      menar      skog      inom      kort
## 0.006840077 0.006982931 0.005790992 0.006091209 0.006346393
##      fraan      stad      loest      fall      vandrar
## 0.006563299 0.006747670 0.005739745 0.006047649 0.005944302
##      soeker      mat      tycker      olika      enkelt
## 0.005844572 0.010817589 0.005766341 0.006070255 0.006328582
##      svara      speciellt      bryr      nere      sett
## 0.006548160 0.006734801 0.006139820 0.006387712 0.006598421
## tillraeckligt      risk      heller      lever      lika

```

```
## 0.005506868 0.005120355 0.005521167 0.005167992 0.005561658
##      laenge      tag      pass
## 0.005896275 0.006180699 0.006422459
##
## $value
## [1] 1
##
## $options
## NULL
```

How to calculate Z-values based on randomized networks

```
#An example of calculating the Z-value for the diameter
#First define the function
Z_d<-function(g,nit){
  d<-vector()
  rew<-10*length(E(g))
  dx<-diameter(g)
  for (i in 1:nit){
    h<-rewire(g,with= keeping_degseq(niter = vcount(g) * 10))
    d[i]<-diameter(h)
  }
  Z<-(dx-mean(d,na.rm=T))/sd(d,na.rm=T)
  X<-dx
  M<-mean(d,na.rm=T)
  SD<-sd(d,na.rm=T)
  result<-data.frame(X,M,SD,Z)
  return(result)
}
#Then call the function (the diameter is the first entry)
Z_d(example_net,100)
```

```
##      X      M      SD      Z
## 1 26 24.12 3.159977 0.594941
```