###############################################################

######################### CHAPTER 10: text2vec ###################

###############################################################

## Example 1 ##

## generated data: analyzing a simple sentence in text2vec

library(text2vec)

x=c("one flew over the nest one flew one flew always sir slave nest the")

txt=x

txt

weights=c(1,1,1) ## equal weighting

it = itoken(txt,tolower,tokenizer = word\_tokenizer, n\_chunks = 1)

vocab <- create\_vocabulary(it, ngram = c(1L,1L))

vectorizer = vocab\_vectorizer(vocab)

tcm=create\_tcm(it, vectorizer, skip\_grams\_window= 3L, skip\_grams\_window\_context="symmetric", weights)

## skip\_grams\_window = 3L indicates that one should look within 3 words to the left

## and 3 words to the right

tcm

word1="flew"

word2="one"

res1=tcm[labels(tcm)[[1]]==word1,labels(tcm)[[2]]==word2]

res2=tcm[labels(tcm)[[1]]==word2,labels(tcm)[[2]]==word1]

res=res1+res2

if(length(res)==0)res=0

res

## Example 2 ##

## toy example: test.csv

## read and clean data

data <- read.csv('C:\\Users\\ledolter\\Desktop\\test.csv', header=FALSE, stringsAsFactors=F)

dim(data)

data[1:4,1]

dim(data)[1] ## number of speeches

for (i in 1:dim(data)[1]) {

txt=data[i,1]

txt=tolower(txt)

txt=gsub("[.]","", ignore.case = TRUE,txt)

txt=gsub("[,]","", ignore.case = TRUE,txt)

txt=gsub("[;]","", ignore.case = TRUE,txt)

txt=gsub("new york","new-york", ignore.case = TRUE,txt) ## the state of new york

data[i,1]=txt

}

data[1:4,1]

## omit meta variables from the text

## speaker in meta2; determine length of each speech

## works if there are no missing values in meta2

len=dim(dim(data)[1])

meta1=dim(dim(data)[1])

meta2=dim(dim(data)[1])

for (i in 1:dim(data)[1]) {

txt=data[i,1]

temp=strsplit(txt, " ")

len[i]=length(temp[[1]])-2

meta1[i]=temp[[1]][1]

meta2[i]=temp[[1]][2]

tempr=dim(len[i])

for (j in 1:len[i]) {

tempr[j]=temp[[1]][j+2]

}

data[i,1]=toString(tempr)

data[i,1]=gsub("[,]","",data[i,1]) ## omitting the comma

}

data[1:4,1]

len

hist(len)

boxplot(len)

quantile(len)

meta2

## omit meta variables from the text

## combining the four speeches

txt=paste(data[1,1],data[2,1],data[3,1],data[4,1],sep=",")

txt

## text2vec Chapter 8

library(text2vec)

weights=c(1,1,1,1,1) ## equal weighting

it = itoken(txt,tolower,tokenizer = word\_tokenizer, n\_chunks = 1)

vocab <- create\_vocabulary(it, ngram = c(1L,1L))

vectorizer = vocab\_vectorizer(vocab)

tcm=create\_tcm(it, vectorizer, skip\_grams\_window= 5L, skip\_grams\_window\_context="symmetric", weights)

## skip\_grams\_window = 3L indicates that one should look within 5 words to the left

## and 5 words to the right

tcm

dim(tcm)

rownames(tcm)

colnames(tcm)

max(tcm)

word1="of"

word2="the"

res1=tcm[labels(tcm)[[1]]==word1,labels(tcm)[[2]]==word2]

res2=tcm[labels(tcm)[[1]]==word2,labels(tcm)[[2]]==word1]

res=res1+res2

if(length(res)==0)res=0

res

word1="purpose"

word2="of"

res1=tcm[labels(tcm)[[1]]==word1,labels(tcm)[[2]]==word2]

res2=tcm[labels(tcm)[[1]]==word2,labels(tcm)[[2]]==word1]

res=res1+res2

if(length(res)==0)res=0

res

uu=as.matrix(tcm)

ind=uu>=1

ind

maxdim=dim(tcm)[1]\*dim(tcm)[2]/2

out1=dim(maxdim)

out2=dim(maxdim)

coocc=dim(maxdim)

k=0

imax=dim(tcm)[1]-1

for (i in 1:imax) {

i1=i+1

for (j in i1:dim(tcm)[2]) {

if (ind[i,j]==TRUE) {

k=k+1

out1[k]=rownames(tcm)[i]

out2[k]=colnames(tcm)[j]

coocc[k]=tcm[i,j]

}

}

}

## summary of co-occurances within a window of length 11 (+/- 5 positions)

wf=data.frame(out1,out2,coocc)

wf[order(coocc,decreasing=TRUE),]

head(wf[order(coocc,decreasing=TRUE),])