Title

Name

date

Let's take a look a set of Dutch speeches in English from the EUSpeech dataset. Use setwd() to set the working directory to the folder that contains Dutch speeches in the file speeches_nl.csv. Read in the speeches and do some cleaning as follows:

```
Sys.setlocale(locale = "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
library(foreign)
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 3.5.2
```

```
library(readtext)
library(topicmodels)

speeches <- read.csv(file = "speeches_nl.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE,
                     sep = ",")

speeches$text <- str_replace_all( speeches$text, "<.+?>", "" )

#we'll set the seed to get reproducible random starting values
set.seed(2)
```

1) Turn the speeches dataframe into a corpus, and then into a dfm object. Remove punctuation and numbers and stopwords, apply stemming, and lowercase the corpus

```
speeches <- corpus(speeches)

speeches.dfm <- dfm(speeches,
                    tolower = TRUE, stem = TRUE,
                    remove=stopwords("english"),
                    remove_punct=TRUE, ngrams = 1,
                    remove_numbers = TRUE)
```

2) Apply a Wordfish scaling model to these speeches.

```
speeches.wordfish <- textmodel_wordfish(speeches.dfm)
```

3) Locate the speeches with the highest score and lowest score on the underlying estimated dimension. Take a look at those two speeches. Do they span an ideological dimension? Write a few sentences on why yes or why no.

```
which.max(speeches.wordfish$theta)
```

```
## [1] 15
```

```r
#memorial speech

which.min(speeches.wordfish$theta)
```

```
## [1] 81
```

```r
#economic ties speech
```

4) Use the `textstat_simil()` function in `quanteda` to calculate the cosine similarity between these two speeches?

```r
textstat_simil(speeches.dfm, c("text15"), method = "cosine", margin = "documents")
```

```
##              text15
## text1   0.13927728
## text2   0.13072112
## text3   0.34414870
## text4   0.22201959
## text5   0.29099036
## text6   0.18093995
## text7   0.40015703
## text8   0.34460402
## text9   0.13072112
## text10  0.25518273
## text11  0.28863072
## text12  0.23349140
## text13  0.27112135
## text14  0.12348792
## text15  1.00000000
## text16  0.07139148
## text17  0.14714689
## text18  0.11142718
## text19  0.11044737
## text20  0.07272779
## text21  0.20679943
## text22  0.20987207
## text23  0.17120957
## text24  0.25665700
## text25  0.16322446
## text26  0.12054864
## text27  0.14326172
## text28  0.08229814
## text29  0.13775969
## text30  0.21678997
## text31  0.09990782
## text32  0.18010420
## text33  0.19707649
## text34  0.36289827
## text35  0.18876833
## text36  0.15728174
## text37  0.12657248
## text38  0.15888920
## text39  0.14454321
## text40  0.12569364
## text41  0.12649225
```

```
## text42  0.25644485
## text43  0.21346981
## text44  0.13656491
## text45  0.09447658
## text46  0.14746377
## text47  0.15741276
## text48  0.16474265
## text49  0.12219708
## text50  0.18630890
## text51  0.16163750
## text52  0.11524287
## text53  0.09355286
## text54  0.10598550
## text55  0.12917788
## text56  0.07485618
## text57  0.16036974
## text58  0.08663201
## text59  0.14729467
## text60  0.05734113
## text61  0.13114267
## text62  0.13101173
## text63  0.10188729
## text64  0.14595128
## text65  0.29031179
## text66  0.27419999
## text67  0.19511896
## text68  0.16665317
## text69  0.06167367
## text70  0.15488555
## text71  0.29089734
## text72  0.18005628
## text73  0.14236295
## text74  0.15731239
## text75  0.09577805
## text76  0.12144177
## text77  0.15411203
## text78  0.16224808
## text79  0.15778100
## text80  0.16929781
## text81  0.12715248
## text82  0.12076147
## text83  0.07707878
## text84  0.25018304
## text85  0.19819424
## text86  0.23732223
## text87  0.19613415
## text88  0.22145109
## text89  0.12076581
## text90  0.12347602
## text91  0.18674731
## text92  0.10982504
## text93  0.09677882
## text94  0.18075725
## text95  0.19735347
```

```
## text96  0.36768971
## text97  0.18465741
## text98  0.16029315
## text99  0.11503946
## text100 0.15605040
## text101 0.13218946
## text102 0.12649225
## text103 0.10063026
## text104 0.11876126
## text105 0.10535320
## text106 0.12980451
## text107 0.16068774
## text108 0.08697969
## text109 0.14692577
## text110 0.05798547
## text111 0.12739219
## text112 0.10317971
## text113 0.14965060
## text114 0.20178303
## text115 0.16516958
## text116 0.13240969
## text117 0.15447512
## text118 0.11787577
## text119 0.18713652
## text120 0.16224808
## text121 0.15732546
## text122 0.17738762
## text123 0.16929781
## text124 0.12715248
## text125 0.12715248
## text126 0.25018304
## text127 0.16650787
## text128 0.23108952
## text129 0.18175113
## text130 0.12059967
## text131 0.13569381
## text132 0.22434009
```

Let's turn to LDA topic models. First, take a sample of 15 speeches as a test set. The rest of our speeches will be our trainig set:

```r
#we have 132 speeches. let's divide them so that we have 117 speeches as our training set, and 15 as ou

docvars(speeches.dfm, "id_numeric") <- 1:ndoc(speeches)
id_train <- sample(1:132, 117, replace = FALSE)

train.dfm <- dfm_subset(speeches.dfm, id_numeric %in% id_train)

#and the 15 remaining speeches as our test data.
test.dfm <- dfm_subset(speeches.dfm, !id_numeric %in% id_train)


#convert both dfms to the topic model library
train.lda.dfm <- convert(train.dfm, to = "topicmodels")
test.lda.dfm <- convert(test.dfm, to = "topicmodels")
```

5) Estimate a 2 topic, a 4 topic and a 6 topic LDA model on `train.dfm` using Gibbs sampling using the LDA function in the `topicmodels` library on a converted `quanteda` dfm object. Call the LDA output `speeches.lda.2`, `speeches.lda.4`, `speeches.lda.6` respectively.

```
speeches.lda.2 <- LDA(train.lda.dfm, method = "Gibbs", k = 2)
speeches.lda.4 <- LDA(train.lda.dfm, method = "Gibbs", k = 4)
speeches.lda.6 <- LDA(train.lda.dfm, method = "Gibbs", k = 6)
```

6) Calculate the perplexity score for each of the 3 models on the `test.lda.dfm` using the `perplexity()` function in the `topicmodels` library. This syntax of the `perplexity()` function is `perplexity(object, newdata)` where `object` is your lda model, and `newdata` your test set.

```
perplexity(object = speeches.lda.2, newdata = test.lda.dfm)
```

```
## [1] 2065.773
```

```
perplexity(object = speeches.lda.4, newdata = test.lda.dfm)
```

```
## [1] 1884.839
```

```
perplexity(object = speeches.lda.6, newdata = test.lda.dfm)
```

```
## [1] 1788.051
```

7) According to the perplexity criterion, which would be the most appropriate topic model?

```
#The 6 topic lda model, since it has the smallest perplexity on the test set
```

8) Inspect the 10 highest loading words on the 6 topic LDA model. How would you interpret each topic? [NB: no right or wrong here]

```
terms(speeches.lda.6, 10)
```

```
##       Topic 1      Topic 2      Topic 3      Topic 4    Topic 5
##  [1,] "dutch"      "netherland" "one"        "world"    "netherland"
##  [2,] "countri"    "europ"      "freedom"    "intern"   "develop"
##  [3,] "netherland" "us"         "today"      "secur"    "year"
##  [4,] "busi"       "trade"      "year"       "countri"  "un"
##  [5,] "trade"      "econom"     "peopl"      "nation"   "women"
##  [6,] "china"      "like"       "live"       "presid"   "work"
##  [7,] "indonesia"  "invest"     "netherland" "right"    "govern"
##  [8,] "can"        "togeth"     "world"      "also"     "achiev"
##  [9,] "also"       "one"        "day"        "peopl"    "dutch"
## [10,] "mani"       "make"       "victim"     "european" "commit"
##       Topic 6
##  [1,] "can"
##  [2,] "sustain"
##  [3,] "like"
##  [4,] "innov"
##  [5,] "need"
##  [6,] "chang"
##  [7,] "new"
##  [8,] "govern"
##  [9,] "work"
## [10,] "climat"
```

9) Write a line of code to count the number of speeches for which topic 6 is the most prevalent topic.

```
sum(topics(speeches.lda.6, 1) == 6)
```

```
## [1] 17
```

10) Write a line of code to display the average topic proportion of each of the 6 topics over all 117 training speeches.

```r
topic.proportions <- posterior(speeches.lda.6)$topics
colMeans(topic.proportions)
```

```
##         1         2         3         4         5         6 
## 0.1974617 0.1603453 0.1830540 0.1497694 0.1431353 0.1662342
```