subsetr

Michael Pearson

3/17/2018

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr, quietly = TRUE)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr, quietly = TRUE)
library(R.utils, quietly = TRUE)
```

```
##
## Attaching package: 'R.oo'
```

```
## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods
```

```
## The following objects are masked from 'package:base':
##
##     attach, detach, gc, load, save
```

```
##
## Attaching package: 'R.utils'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, inherits, isOpen, parse, warnings
```

```
library(SnowballC, quietly = TRUE)
library(tidyr, quietly = TRUE)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:R.utils':
##
##     extract
```

```
library(data.table, quietly = TRUE)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 3.4.4
```

```
## Package version: 1.2.0
```

```
## Parallel computing: 2 of 4 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':
##
##     View
```

```
library(data.table, quietly = TRUE)
library(readtext)
library(stringr)
```

Remove the one-offs

now let's process the ones with multiple bigrams

```
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 downstream <- 0.5
 for (j in 1:b)
  {
    mid_tri <- data.table()
    if(nrow(tim) - a >= c )
      {
setkey(trixy,word1)
      for (i in d:a)
{
##setkey(trixy,bigrams)
tardis <- trixy[as.character(aggy$word1[i])]
tardis$prob <- (tardis$bi_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
##trixy <- trixy[bigrams != aggy$bigrams[i],]
##print(paste("i is ",i))
##print(paste("number of rows in trixy is ",nrow(trixy)))
}
    d <- a + 1
    a <- a + c
      }
    else {
```

```
      a <- nrow(tim)
      d <- 100*floor(nrow(tim)/100) + 1
    for (i in d:a)
{
tardis <- trixy[word1 == aggy$word1[i],]
tardis$prob <- (tardis$bi_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_bi_ns_ns <- read.csv("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Pr
combi_bi_ns_ns <- data.table(combi_bi_ns_ns)
trixy <- combi_bi_ns_ns[combi_bi_ns_ns$bi_gram_ns_ns >= 4,]
aggy <- trixy[,.(sum = sum(bi_gram_ns_ns)), by = word1]
aggy <- aggy[aggy$sum >= 10]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project
rm(trixy)
rm(aggy)
rm(combi_bi_ns_ns)
rm(blah)
```

Now the Trigrams

```
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/1000)
b <- 1001
c <- a
d <- 1
 full_tri <- data.table()
 downstream <- 0.5
 for (j in 1:b)
  {
    mid_tri <- data.table()
   if(nrow(tim) - a >= c )
      {
setkey(trixy,bigrams)
      for (i in d:a)
{
tardis <- trixy[as.character(aggy$bigrams[i])]
tardis$prob <- (tardis$tri_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    d <- a + 1
    a <- a + c
      }
    else {
      a <- nrow(tim)
```

```
      d <- 1000*floor(nrow(tim)/1000) + 1
    for (i in d:a)
{
tardis <- trixy[bigrams == aggy$bigrams[i],]
tardis$prob <- (tardis$tri_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
##trixy <- trixy[bigrams != aggy$bigrams[i],]
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_tri_ns_ns <- read.csv("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone P
combi_tri_ns_ns <- data.table(combi_tri_ns_ns)
trixy <- combi_tri_ns_ns[combi_tri_ns_ns$tri_gram_ns_ns >= 4,]
aggy <- trixy[,.(sum = sum(tri_gram_ns_ns)), by = bigrams]
aggy <- aggy[aggy$sum >= 10]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project/
rm(trixy)
rm(aggy)
rm(combi_tri_ns_ns)
rm(blah)
```

should run first

```
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 downstream <- 0.5
 for (j in 1:b)
  {
    mid_tri <- data.table()
    if(nrow(tim) - a >= c )
      {
setkey(trixy,trigrams)
      for (i in d:a)
{
tardis <- trixy[as.character(aggy$trigrams[i])]
tardis$prob <- (tardis$quad_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    d <- a + 1
    a <- a + c
      }
    else {
      a <- nrow(tim)
```

```
      d <- 100*floor(nrow(tim)/100) + 1
    for (i in d:a)
{
tardis <- trixy[as.character(aggy$trigrams[i])]
tardis$prob <- (tardis$tri_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_quad_ns_ns <- read.csv("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
combi_quad_ns_ns <- data.table(combi_quad_ns_ns)
trixy <- combi_quad_ns_ns[combi_quad_ns_ns$quad_gram_ns_ns >= 4,]
aggy <- trixy[,.(sum = sum(quad_gram_ns_ns)), by = trigrams]
aggy <- aggy[aggy$sum >= 10]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project/
rm(trixy)
rm(aggy)
rm(combi_quad_ns_ns)
rm(blah)
```

Now the Quin-grams

```
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 downstream <- 0.5
 for (j in 1:b)
  {
    mid_tri <- data.table()
    if(nrow(tim) - a >= c )
      {
setkey(trixy,quadgrams)
      for (i in d:a)
{
tardis <- trixy[as.character(aggy$quadgrams[i])]
tardis$prob <- (tardis$quin_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    d <- a + 1
    a <- a + c
      }
    else {
      a <- nrow(tim)
      d <-  d <- 100*floor(nrow(tim)/100) + 1
    for (i in d:a)
{
```

```
tardis <- trixy[as.character(aggy$trigrams[i])]
tardis$prob <- (tardis$quad_gram_ns_ns - downstream)/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_quin_ns_ns <- read.csv("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
combi_quin_ns_ns <- data.table(combi_quin_ns_ns)
trixy <- combi_quin_ns_ns[combi_quin_ns_ns$quin_gram_ns_ns >= 4,]
aggy <- trixy[,.(sum = sum(quin_gram_ns_ns)), by = quadgrams]
aggy <- aggy[aggy$sum >= 10]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project/
rm(trixy)
rm(aggy)
rm(combi_quin_ns_ns)
rm(blah)
```