```
---
title: "Thirdprocess"
author: "Michael Pearson"
date: "8/19/2020"
output:
  pdf_document: default
  word_document: default
  html_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(dplyr, quietly = TRUE)
library(readr, quietly = TRUE)
#library(R.utils, quietly = TRUE)
#library(SnowballC, quietly = TRUE)
library(tidyr, quietly = TRUE)
library(data.table, quietly = TRUE)
#library(quanteda)
library(stringr)
#library(tinytex)
```

Remove the one-offs

now let's process the ones with multiple bigrams
```{r bigrams, eval = TRUE}
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 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$bi_gram_ns_ns/aggy$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$bi_gram_ns_ns/aggy$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/Coursera/Capstone
Project/20sample/bi_gram_ns_ns.csv", colClasses = c( NA, NA, NA) )
combi_bi_ns_ns <- data.table(combi_bi_ns_ns)
trixy <- combi_bi_ns_ns[combi_bi_ns_ns$bi_gram_ns_ns >= 2,]
##trixy <- data.table(combi_bi_ns_ns)
aggy <- trixy[,.(sum = sum(bi_gram_ns_ns)), by = word1]
aggy <- aggy[aggy$sum >= 70]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/nosingles_bi_ns_ns.csv" )
rm(trixy)
rm(aggy)
rm(combi_bi_ns_ns)
rm(blah)
##print(traa)
```
Now the Trigrams

``` {r trigrams, eval = TRUE}
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/1000)
b <- 1001
c <- a
d <- 1
 full_tri <- data.table()
 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$tri_gram_ns_ns/aggy$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$tri_gram_ns_ns/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
##trixy <- trixy[bigrams != aggy$bigrams[i],]
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_tri_ns_ns <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/combi_tri_ns_ns.csv", colClasses = c("NULL", NA, NA, NA) )
combi_tri_ns_ns <- data.table(combi_tri_ns_ns)
trixy <- combi_tri_ns_ns[combi_tri_ns_ns$tri_gram_ns_ns >= 2,]
##trixy <- data.table(combi_tri_ns_ns)
aggy <- trixy[,.(sum = sum(tri_gram_ns_ns)), by = bigrams]
aggy <- aggy[aggy$sum >= 50]
aggy <- data.table(aggy)
traa <- system.time(blocky(trixy, aggy, full_tri))
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/nosingles_tri_ns_ns.csv" )
##rm(trixy)
##rm(aggy)
##rm(combi_tri_ns_ns)
##rm(blah)
print(traa)
```

should run first


``` {r quadgrams, eval = TRUE}
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 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$quad_gram_ns_ns/aggy$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$tri_gram_ns_ns/aggy$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/Coursera/Capstone
Project/20sample/combi_quad_ns_ns.csv", colClasses = c("NULL", NA, NA, NA)
)
combi_quad_ns_ns <- data.table(combi_quad_ns_ns)
trixy <- combi_quad_ns_ns[combi_quad_ns_ns$quad_gram_ns_ns >= 2,]
##trixy <- data.table(combi_quad_ns_ns)
aggy <- trixy[,.(sum = sum(quad_gram_ns_ns)), by = trigrams]
aggy <- aggy[aggy$sum >= 6]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/nosingles_quad_ns_ns.csv" )
rm(trixy)
rm(aggy)
rm(combi_quad_ns_ns)
rm(blah)
```
Now the Quin-grams

```{r quingrams, eval = TRUE}
blocky <- function(trap, tim, ful_tri) {
a <- floor(nrow(tim)/100)
b <- 101
c <- a
d <- 1
 full_tri <- data.table()
 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/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/aggy$sum[i]
mid_tri <- rbind(mid_tri, tardis)
}
    }
   full_tri <- rbind(full_tri, mid_tri)
 }
return(full_tri)
}
combi_quin_ns_ns <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/combi_quin_ns_ns.csv", colClasses = c("NULL", NA, NA, NA)
)
combi_quin_ns_ns <- data.table(combi_quin_ns_ns)
trixy <- combi_quin_ns_ns[combi_quin_ns_ns$quin_gram_ns_ns >= 1,]
##trixy <- data.table(combi_quin_ns_ns)
aggy <- trixy[,.(sum = sum(quin_gram_ns_ns)), by = quadgrams]
aggy <- aggy[aggy$sum >= 3]
aggy <- data.table(aggy)
blah <- blocky(trixy, aggy, full_tri)
write.csv(blah,file = "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/nosingles_quin_ns_ns.csv" )
rm(trixy)
rm(aggy)
rm(combi_quin_ns_ns)
rm(blah)
```