Singlefilegrams

Michael Pearson

6/27/2018

Will read an input and search for n-grams

Start with bigrams

```
start2 <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone Project/20sample/nosingles_bi_ns_ns.cs
start2 <- data.table(start2)
topn <- function(input, maxn) {
countem <- input[,.N,by=word1]
countem <- data.table(countem)
a <- nrow(countem)
    mid_tri <- data.table()
 for (j in 1:a)
  {
    if(countem$N[j] > maxn)
      {
setkey(input,word1)
      intermediate <- input[as.character(countem$word1[j])]
      nother <- intermediate[1:5,]
      mid_tri <- rbind(mid_tri, nother)
}
    else {
   setkey(input,word1)
      intermediate <- input[as.character(countem$word1[j])]
      mid_tri <- rbind(mid_tri, intermediate)
    }
 }
return(mid_tri)
 }

    start2 <- topn(start2,5)
trap <- start2[,c(1,2,4)]
trap <- cbind(c(2),trap)
colnames(trap)[1] <- c("n")
colnames(trap)[2] <-c("n-gram")
colnames(trap)[3] <-c("next word")
rm(start2)
```

Go to tri-grams

```
start3 <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone Project/20sample/nosingles_tri_ns_ns.
start3 <- data.table(start3)
topn <- function(input, maxn) {
countem <- input[,.N,by=bigrams]
countem <- data.table(countem)
a <- nrow(countem)
    mid_tri <- data.table()
 for (j in 1:a)
  {
    if(countem$N[j] > maxn)
```

```
      {
setkey(input,bigrams)
      intermediate <- input[as.character(countem$bigrams[j])]
      nother <- intermediate[1:5,]
      mid_tri <- rbind(mid_tri, nother)
}
    else {
   setkey(input,bigrams)
      intermediate <- input[as.character(countem$bigrams[j])]
      mid_tri <- rbind(mid_tri, intermediate)
    }
 }
return(mid_tri)
 }
start3 <- topn(start3,5)
trap3 <- start3[,c(1,2,4)]
trap3 <- cbind(c(3),trap3)
colnames(trap3)[1] <- c("n")
colnames(trap3)[2] <-c("n-gram")
colnames(trap3)[3] <-c("next word")
trap <- rbind(trap, trap3)
rm(trap3)
rm(start3)
```

Now the quad grams

```
start4 <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone Project/20sample/nosingles_quad_ns_ns
start4 <- data.table(start4)
topn <- function(input, maxn) {
countem <- input[,.N,by=trigrams]
countem <- data.table(countem)
a <- nrow(countem)
    mid_tri <- data.table()
 for (j in 1:a)
  {
    if(countem$N[j] > maxn)
      {
setkey(input,trigrams)
      intermediate <- input[as.character(countem$trigrams[j])]
      nother <- intermediate[1:5,]
      mid_tri <- rbind(mid_tri, nother)
}
    else {
   setkey(input,trigrams)
      intermediate <- input[as.character(countem$trigrams[j])]
      mid_tri <- rbind(mid_tri, intermediate)
    }
 }
return(mid_tri)
}
start4 <- topn(start4, 5)
trap4 <- start4[,c(1,2,4)]
trap4 <- cbind(c(4),trap4)
colnames(trap4)[1] <- c("n")
```

```
colnames(trap4)[2] <-c("n-gram")
colnames(trap4)[3] <-c("next word")
trap <- rbind(trap, trap4)
rm(trap4)
rm(start4)
```

And lastly the quins,

then write the file

```
start5 <- read.csv("/Users/mutecypher/Documents/Coursera/Capstone Project/20sample/nosingles_quin_ns_ns
start5 <- data.table(start5)
topn <- function(input, maxn) {
countem <- input[,.N,by=quadgrams]
countem <- data.table(countem)
a <- nrow(countem)
    mid_tri <- data.table()
 for (j in 1:a)
  {
    if(countem$N[j] > maxn)
      {
setkey(input,quadgrams)
      intermediate <- input[as.character(countem$quadgrams[j])]
      nother <- intermediate[1:5,]
      mid_tri <- rbind(mid_tri, nother)
}
    else {
   setkey(input,quadgrams)
      intermediate <- input[as.character(countem$quadgrams[j])]
      mid_tri <- rbind(mid_tri, intermediate)
    }
 }
return(mid_tri)
}
start5 <- topn(start5, 5)
trap5 <- start5[,c(1,2,4)]
trap5 <- cbind(c(5),trap5)
colnames(trap5)[1] <- c("n")
colnames(trap5)[2] <-c("n-gram")
colnames(trap5)[3] <-c("next word")
trap <- rbind(trap, trap5)
rm(trap5)
rm(start5)
write.csv(trap,file = "/Users/mutecypher/Documents/Coursera/Capstone Project/20sample/allgrams_ns_ns.csv
```