~~~
---
title: "Firstprocess"
author: "Michael Pearson"
date: "8/17/2020"
output:
  pdf_document: default
  word_document: default
---

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


## Quanteda work
This will create a corpus, clean it, and tokenize it using quanteda

```{r basic input}
## count the lines in the twitter, news, and blog files
newslines <- countLines("/Users/mutecypher/Documents/Coursera/Capstone
Project/files/en_US/en_US.news.txt")
bloglines <- countLines("/Users/mutecypher/Documents/Coursera/Capstone
Project/files/en_US/en_US.blogs.txt")
tweetlines <- countLines("/Users/mutecypher/Documents/Coursera/Capstone
Project/files/en_US/en_US.twitter.txt")
## use that to read the files
tweet_us <- file("/Users/mutecypher/Documents/Coursera/Capstone Project/
files/en_US/en_US.twitter.txt")
tweet_all <- readLines(tweet_us, n= tweetlines, warn = FALSE, encoding =
"UTF=8", skipNul = TRUE)
close(tweet_us)
blog_us <- file("/Users/mutecypher/Documents/Coursera/Capstone Project/
files/en_US/en_US.blogs.txt")
blog_all <- readLines(blog_us, n= bloglines, warn = FALSE, encoding =
"UTF=8", skipNul = TRUE)
close(blog_us)
news_us <- file("/Users/mutecypher/Documents/Coursera/Capstone Project/
files/en_US/en_US.news.txt")
news_all <- readLines(news_us, n = newslines, warn = FALSE, encoding =
"UTF=8", skipNul = TRUE)
close(news_us)
```

## Sample 20% of the files to get a test sample corpus

```{r build the sample}
~~~

~~~~
set.seed(8172020)
samp_per <- 0.20
sam_twit <-
tweet_all[sample(1:length(tweet_all),samp_per*length(tweet_all), replace =
FALSE)]
sam_test <- tweet_all[-
sample(1:length(tweet_all),samp_per*length(tweet_all), replace = FALSE)]
write_lines(sam_twit, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/samples/twittersample.txt")
write_lines(sam_test, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/test/twittertest.txt")
sam_news <- news_all[sample(1:length(news_all),samp_per*length(news_all))]
news_test <- news_all[-
sample(1:length(news_all),samp_per*length(news_all))]
write_lines(news_test, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/test/newstest.txt")
write_lines(sam_news, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/samples/newssample.txt")
sam_blog <- blog_all[sample(1:length(blog_all),samp_per*length(blog_all))]
blog_test <- blog_all[-
sample(1:length(blog_all),samp_per*length(blog_all))]
write_lines(sam_blog, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/samples/blogsample.txt")
write_lines(blog_test, "/Users/mutecypher/Documents/Coursera/Capstone
Project/files/test/blogtest.txt")
samp <- "/Users/mutecypher/Documents/Coursera/Capstone Project/files/
samples/"
samplename <- readtext(samp)
myCorpus <- corpus(samplename)
test_name <- "/Users/mutecypher/Documents/Coursera/Capstone Project/files/
test/"
testname <- readtext(test_name)
testCorpus <- corpus(testname)
```
## Now make the n-grams - with and without stems
``` {r make the n-grams, eval = TRUE}
## onegrams with stemming and stopwords
one_gram <- tokens(myCorpus, what = "word", remove_numbers = TRUE,
remove_punct = TRUE, remove_symbols = TRUE, split_hyphens = TRUE,
remove_url = TRUE)
ns_one_gram <- tokens_remove(one_gram, stopwords("english"))
dfm_one_gram_stem_and_stop <- dfm(ns_one_gram, tolower = TRUE, stem = TRUE)
one_gram_s_s <- sort(colSums(dfm_one_gram_stem_and_stop), decreasing =
TRUE)
one_gram_s_s <- data.frame(one_gram_s_s)
one_gram_s_s <- setDT(one_gram_s_s, keep.rownames = TRUE)
write_csv(one_gram_s_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/one_gram_s_s.csv")
rm(dfm_one_gram_stem_and_stop)
rm(one_gram_s_s)
```

## test onegram - no stemming, but stopwords kept - all from testCorpus
```{r one grams for test, eval = TRUE}
~~~~

~~~
## I don't recall what this does
test_one_gram <- tokens(testCorpus, what = "word", remove_numbers = TRUE,
remove_punct = TRUE, remove_symbols = TRUE, split_hyphens = TRUE,
remove_url = TRUE)
ts_one_gram <- tokens_remove(test_one_gram, stopwords("english"))
dfm_one_gram_test <- dfm(ts_one_gram, tolower = TRUE, stem = FALSE)
one_gram_test <- sort(colSums(dfm_one_gram_test), decreasing = TRUE)
one_gram_test <- data.frame(one_gram_test)
one_gram_test <- setDT(one_gram_test, keep.rownames = TRUE)
write_csv(one_gram_test, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20test/test_one_gram.csv")
rm(dfm_one_gram_test)
rm(one_gram_test)
rm(ts_one_gram)
rm(test_one_gram)
```
## onegram with no stemming, but yes to stop words
```{r no stemming but stopping, eval= TRUE}
dfm_one_gram_nostem_and_stop <- dfm(one_gram, tolower = TRUE, stem = FALSE)
one_gram_ns_s <- sort(colSums(dfm_one_gram_nostem_and_stop), decreasing =
TRUE)
one_gram_ns_s <- data.frame(one_gram_ns_s)
one_gram_ns_s <- setDT(one_gram_ns_s, keep.rownames = TRUE)
write_csv(one_gram_ns_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/one_gram_ns_s.csv")
rm(one_gram_ns_s)
rm(dfm_one_gram_nostem_and_stop)
```
## onegram with stemming on no stopwords
``` {r onegram with stemming on no stopwords, eval = TRUE}
dfm_one_gram_stem_and_nostop <- dfm(one_gram, tolower = TRUE, stem = TRUE)
one_gram_s_ns <- sort(colSums(dfm_one_gram_stem_and_nostop), decreasing =
TRUE)
one_gram_s_ns <- data.frame(one_gram_s_ns)
one_gram_s_ns <- setDT(one_gram_s_ns, keep.rownames = TRUE)
write_csv(one_gram_s_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/one_gram_s_ns.csv")
rm(one_gram_s_ns)
rm(dfm_one_gram_stem_and_nostop)
```
## no stemming or stopwords

``` {r no stemming or stopwords, eval = TRUE}
dfm_one_gram_nostem_and_nostop <- dfm(one_gram, tolower = TRUE, stem =
FALSE)
one_gram_ns_ns <- sort(colSums(dfm_one_gram_nostem_and_nostop), decreasing
= TRUE)
one_gram_ns_ns <- data.frame(one_gram_ns_ns)
one_gram_ns_ns <- setDT(one_gram_ns_ns, keep.rownames = TRUE)
write_csv(one_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/one_gram_ns_ns.csv")
rm(one_gram_ns_ns)
rm(dfm_one_gram_nostem_and_nostop)
rm(one_gram)
~~~

~~~
```


## Now we will do a bunch of bi_grams

``` {r bigrams with stemming and stop words removed, eval = TRUE}
##bigrams with stemming and stop words removed
bi_gram <- tokens(myCorpus, remove_numbers = TRUE, remove_punct = TRUE,
remove_symbols = TRUE, what = "word", split_hyphens = TRUE, remove_url =
TRUE, ngrams = 2L, concatenator = " ")
dfm_bi_gram_stem_stop <- dfm(bi_gram, tolower = TRUE, stem = TRUE, remove =
stopwords("english"))
bi_gram_s_s <- sort(colSums(dfm_bi_gram_stem_stop), decreasing = TRUE)
bi_gram_s_s <- data.frame(bi_gram_s_s)
bi_gram_s_s <- setDT(bi_gram_s_s, keep.rownames = TRUE)
bi_gram_s_s <- separate(bi_gram_s_s, rn, c("word1", "word2"), sep = " ")
write_csv(bi_gram_s_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/bi_gram_s_s.csv")
rm(dfm_bi_gram_stem_stop)
rm(bi_gram_s_s)
```
##bigrams with no stemming but stopwords
```{r bigrams with no stemming but stopwords, eval = TRUE}
dfm_bi_gram_nostem_stop <- dfm(bi_gram, tolower = TRUE, stem = FALSE,
remove = stopwords("english"))
bi_gram_ns_s <- sort(colSums(dfm_bi_gram_nostem_stop), decreasing = TRUE)
bi_gram_ns_s <- data.frame(bi_gram_ns_s)
bi_gram_ns_s <- setDT(bi_gram_ns_s, keep.rownames = TRUE)
bi_gram_ns_s <- separate(bi_gram_ns_s, rn, c("word1", "word2"), sep = " ")
write_csv(bi_gram_ns_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/bi_gram_ns_s.csv")
rm(dfm_bi_gram_nostem_stop)
rm(bi_gram_ns_s)
rm(ns_bi_gram)
```
## stemming, but no stopwords
```{r bigrams with stemming but no stopwords, eval = TRUE}
dfm_bi_gram_stem_nostop <- dfm(bi_gram, tolower = TRUE, stem = TRUE)
bi_gram_s_ns <- sort(colSums(dfm_bi_gram_stem_nostop), decreasing = TRUE)
bi_gram_s_ns <- data.frame(bi_gram_s_ns)
bi_gram_s_ns <- setDT(bi_gram_s_ns, keep.rownames = TRUE)
bi_gram_s_ns <- separate(bi_gram_s_ns, rn, c("word1", "word2"), sep = " ")
write_csv(bi_gram_s_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/bi_gram_s_ns.csv")
rm(dfm_bi_gram_stem_nostop)
rm(bi_gram_s_ns)
```

## neither stemming nor stop words

``` {r bigrams with neither stemming nor stop words, eval = TRUE}
dfm_bi_gram_nostem_nostop <- dfm(bi_gram, tolower = TRUE, stem = FALSE)
bi_gram_ns_ns <- sort(colSums(dfm_bi_gram_nostem_nostop), decreasing =
TRUE)
~~~

~~~
bi_gram_ns_ns <- data.frame(bi_gram_ns_ns)
bi_gram_ns_ns <- setDT(bi_gram_ns_ns, keep.rownames = TRUE)
bi_gram_ns_ns <- separate(bi_gram_ns_ns, rn, c("word1", "word2"), sep = "
")
write_csv(bi_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/bi_gram_ns_ns.csv")
rm(dfm_bi_gram_nostem_nostop)
rm(bi_gram_ns_ns)
rm(bi_gram)
```


## now for the test set of bi_grams

```{r bigram test set, eval = TRUE}
bi_gram_test <- tokens(testCorpus, remove_numbers = TRUE, remove_punct =
TRUE, remove_symbols = TRUE, what = "word", split_hyphens = TRUE,
remove_url = TRUE, ngrams = 2L, concatenator = " ")
dfm_bi_gram_nostem_nostop <- dfm(bi_gram_test, tolower = TRUE, stem =
FALSE)
bi_gram_ns_ns <- sort(colSums(dfm_bi_gram_nostem_nostop), decreasing =
TRUE)
bi_gram_ns_ns <- data.frame(bi_gram_ns_ns)
bi_gram_ns_ns <- setDT(bi_gram_ns_ns, keep.rownames = TRUE)
bi_gram_ns_ns <- separate(bi_gram_ns_ns, rn, c("word1", "word2"), sep = "
")
write_csv(bi_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20test/bi_gram_test.csv")
rm(dfm_bi_gram_nostem_nostop)
rm(bi_gram_ns_ns)
rm(bi_gram_test)


```

## trigrams


``` {r trigrams with stemming and stop words removed, eval = TRUE}
##trigrams with stemming and stop words removed
tri_gram <- tokens(myCorpus, remove_numbers = TRUE, remove_punct = TRUE,
remove_symbols = TRUE, what = "word", split_hyphens = TRUE, remove_url =
TRUE, ngrams = 3L,concatenator = " ")
dfm_tri_gram_stem_stop <- dfm(tri_gram, tolower = TRUE, stem = TRUE, remove
= stopwords("english"))
tri_gram_s_s <- sort(colSums(dfm_tri_gram_stem_stop), decreasing = TRUE)
tri_gram_s_s <- data.frame(tri_gram_s_s)
tri_gram_s_s <- setDT(tri_gram_s_s, keep.rownames = TRUE)
tri_gram_s_s <- separate(tri_gram_s_s, rn, c("word1", "word2", "word3"),
sep = " ")
write_csv(tri_gram_s_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/tri_gram_s_s.csv")
rm(dfm_tri_gram_stem_stop)
rm(tri_gram_s_s)
~~~

~~~~
```
##trigrams with no stemming but stopwords
```{r trigrams with no stemming but stopwords, eval = TRUE}
dfm_tri_gram_nostem_stop <- dfm(tri_gram, tolower = TRUE, stem = FALSE,
remove = stopwords("english"))
tri_gram_ns_s <- sort(colSums(dfm_tri_gram_nostem_stop), decreasing = TRUE)
tri_gram_ns_s <- data.frame(tri_gram_ns_s)
tri_gram_ns_s <- setDT(tri_gram_ns_s, keep.rownames = TRUE)
tri_gram_ns_s <- separate(tri_gram_ns_s, rn, c("word1", "word2", "word3"),
sep = " ")
write_csv(tri_gram_ns_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/tri_gram_ns_s.csv")
rm(dfm_tri_gram_nostem_stop)
rm(tri_gram_ns_s)
rm(ns_tri_gram)
```
## stemming, but no stopwords
```{r trigrams with stemming but no stopwords, eval = TRUE}
dfm_tri_gram_stem_nostop <- dfm(tri_gram, tolower = TRUE, stem = TRUE)
tri_gram_s_ns <- sort(colSums(dfm_tri_gram_stem_nostop), decreasing = TRUE)
tri_gram_s_ns <- data.frame(tri_gram_s_ns)
tri_gram_s_ns <- setDT(tri_gram_s_ns, keep.rownames = TRUE)
tri_gram_s_ns <- separate(tri_gram_s_ns, rn, c("word1", "word2", "word3"),
sep = " ")
write_csv(tri_gram_s_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/tri_gram_s_ns.csv")
rm(dfm_tri_gram_nostem_stop)
rm(tri_gram_s_ns)
```

## neither stemming nor stop words

``` {r tri neither stemming nor stop words, eval = TRUE}
dfm_tri_gram_nostem_nostop <- dfm(tri_gram, tolower = TRUE, stem = FALSE)
tri_gram_ns_ns <- sort(colSums(dfm_tri_gram_nostem_nostop), decreasing =
TRUE)
tri_gram_ns_ns <- data.frame(tri_gram_ns_ns)
tri_gram_ns_ns <- setDT(tri_gram_ns_ns, keep.rownames = TRUE)
tri_gram_ns_ns <- separate(tri_gram_ns_ns, rn, c("word1", "word2",
"word3"), sep = " ")
write_csv(tri_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/tri_gram_ns_ns.csv")
rm(dfm_tri_gram_nostem_nostop)
rm(tri_gram_ns_ns)
rm(tri_gram)
```

## test version of tri-grams

```{r test version trigrams, eval = TRUE}
tri_gram_test <- tokens(testCorpus, remove_numbers = TRUE, remove_punct =
TRUE, remove_symbols = TRUE, what = "word", split_hyphens = TRUE,
remove_url = TRUE, ngrams = 3L,concatenator = " ")
~~~~

~~~
dfm_tri_gram_nostem_nostop <- dfm(tri_gram_test, tolower = TRUE, stem =
FALSE)
tri_gram_ns_ns <- sort(colSums(dfm_tri_gram_nostem_nostop), decreasing =
TRUE)
tri_gram_ns_ns <- data.frame(tri_gram_ns_ns)
tri_gram_ns_ns <- setDT(tri_gram_ns_ns, keep.rownames = TRUE)
tri_gram_ns_ns <- separate(tri_gram_ns_ns, rn, c("word1", "word2",
"word3"), sep = " ")
write_csv(tri_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20test/tri_gram_test.csv")
rm(dfm_tri_gram_nostem_nostop)
rm(tri_gram_ns_ns)
rm(tri_gram_test)
```

#quad grams

```{r quadgrams with stemming and stop words removed, eval = TRUE}
##quadgrams with stemming and stop words removed
quad_gram <- tokens(myCorpus, remove_numbers = TRUE, remove_punct = TRUE,
remove_symbols = TRUE, what = "word", split_hyphens = TRUE, remove_url =
TRUE, ngrams = 4L, concatenator = " ")
dfm_quad_gram_stem_stop <- dfm(quad_gram, tolower = TRUE, stem = TRUE,
remove = stopwords("english"))
quad_gram_s_s <- sort(colSums(dfm_quad_gram_stem_stop), decreasing = TRUE)
quad_gram_s_s <- data.frame(quad_gram_s_s)
quad_gram_s_s <- setDT(quad_gram_s_s, keep.rownames = TRUE)
quad_gram_s_s <- separate(quad_gram_s_s, rn, c("word1", "word2", "word3",
"word4"), sep = " ")
write_csv(quad_gram_s_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quad_gram_s_s.csv")
rm(dfm_quad_gram_stem)
rm(quad_gram_s_s)
```
##quadgrams with no stemming but stopwords
```{r quadgrams with no stemming but stopwords, eval = TRUE}
dfm_quad_gram_nostem_stop <- dfm(quad_gram , tolower = TRUE, stem = FALSE,
remove = stopwords("english"))
quad_gram_ns_s <- sort(colSums(dfm_quad_gram_nostem_stop), decreasing =
TRUE)
quad_gram_ns_s <- data.frame(quad_gram_ns_s)
quad_gram_ns_s <- setDT(quad_gram_ns_s, keep.rownames = TRUE)
quad_gram_ns_s <- separate(quad_gram_ns_s, rn, c("word1", "word2", "word3",
"word4"), sep = " ")
write_csv(quad_gram_ns_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quad_gram_ns_s.csv")
rm(dfm_quad_gram_nostem_stop)
rm(quad_gram_ns_s)
```
## stemming, but no stopwords
```{r quadgrams with stemming but no stopwords, eval = TRUE}
dfm_quad_gram_stem_nostop <- dfm(quad_gram, tolower = TRUE, stem = TRUE)
quad_gram_s_ns <- sort(colSums(dfm_quad_gram_stem_nostop), decreasing =
TRUE)
~~~

~~~
quad_gram_s_ns <- data.frame(quad_gram_s_ns)
quad_gram_s_ns <- setDT(quad_gram_s_ns, keep.rownames = TRUE)
quad_gram_s_ns <- separate(quad_gram_s_ns, rn, c("word1", "word2", "word3",
"word4"), sep = " ")
write_csv(quad_gram_s_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quad_gram_s_ns.csv")
rm(dfm_quad_gram_nostem_stop)
rm(quad_gram_s_ns)
```

## neither stemming nor stop words

``` {r quadgrams with neither stemming nor stop words, eval = TRUE}
dfm_quad_gram_nostem_nostop <- dfm(quad_gram, tolower = TRUE, stem = FALSE)
quad_gram_ns_ns <- sort(colSums(dfm_quad_gram_nostem_nostop), decreasing =
TRUE)
quad_gram_ns_ns <- data.frame(quad_gram_ns_ns)
quad_gram_ns_ns <- setDT(quad_gram_ns_ns, keep.rownames = TRUE)
quad_gram_ns_ns <- separate(quad_gram_ns_ns, rn, c("word1", "word2",
"word3", "word4"), sep = " ")
write_csv(quad_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quad_gram_ns_ns.csv")
rm(dfm_quad_gram_nostem_nostop)
rm(quad_gram_ns_ns)
rm(quad_gram)
```


## Now for the test data from Test Corpus of quadgrams

```{r test quadgrams, eval = TRUE}
quad_gram_test <- tokens(testCorpus, remove_numbers = TRUE, remove_punct =
TRUE, remove_symbols = TRUE, what = "word", split_hyphens = TRUE,
remove_url = TRUE, ngrams = 4L, concatenator = " ")
dfm_quad_gram_nostem_nostop <- dfm(quad_gram_test, tolower = TRUE, stem =
FALSE)
quad_gram_ns_ns <- sort(colSums(dfm_quad_gram_nostem_nostop), decreasing =
TRUE)
quad_gram_ns_ns <- data.frame(quad_gram_ns_ns)
quad_gram_ns_ns <- setDT(quad_gram_ns_ns, keep.rownames = TRUE)
quad_gram_ns_ns <- separate(quad_gram_ns_ns, rn, c("word1", "word2",
"word3", "word4"), sep = " ")
write_csv(quad_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20test/quad_gram_test.csv")
rm(dfm_quad_gram_nostem_nostop)
rm(quad_gram_ns_ns)
rm(quad_gram_test)
```

## quin grams


```{r quingrams with stemming and stop words removed, eval = TRUE}
##quingrams with stemming and stop words removed
~~~

~~~
quin_gram <- tokens(myCorpus, remove_numbers = TRUE, remove_punct = TRUE,
remove_symbols = TRUE, what = "word", split_hyphens = TRUE, remove_url =
TRUE, ngrams = 5L,concatenator = " ")
dfm_quin_gram_stem_stop <- dfm(quin_gram , tolower = TRUE, stem = TRUE,
remove = stopwords("english"))
quin_gram_s_s <- sort(colSums(dfm_quin_gram_stem_stop), decreasing = TRUE)
quin_gram_s_s <- data.frame(quin_gram_s_s)
quin_gram_s_s <- setDT(quin_gram_s_s, keep.rownames = TRUE)
quin_gram_s_s <- separate(quin_gram_s_s, rn, c("word1", "word2", "word3",
"word4", "word5"), sep = " ")
write_csv(quin_gram_s_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quin_gram_s_s.csv")
rm(dfm_quin_gram_stem_stop)
rm(quin_gram_s_s)
```
##quingrams with no stemming but stopwords
```{r quingrams with no stemming but stopwords, eval = TRUE}
dfm_quin_gram_nostem_stop <- dfm(quin_gram , tolower = TRUE, stem = FALSE,
remove = stopwords("english"))
quin_gram_ns_s <- sort(colSums(dfm_quin_gram_nostem_stop), decreasing =
TRUE)
quin_gram_ns_s <- data.frame(quin_gram_ns_s)
quin_gram_ns_s <- setDT(quin_gram_ns_s, keep.rownames = TRUE)
quin_gram_ns_s <- separate(quin_gram_ns_s, rn, c("word1", "word2", "word3",
"word4", "word5"), sep = " ")
write_csv(quin_gram_ns_s, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quin_gram_ns_s.csv")
rm(dfm_quin_gram_nostem_stop)
rm(quin_gram_ns_s)
rm(ns_quin_gram)
```
## stemming, but no stopwords
```{r quingrams with stemming but no stopwords, eval = TRUE}
dfm_quin_gram_stem_nostop <- dfm(quin_gram, tolower = TRUE, stem = TRUE)
quin_gram_s_ns <- sort(colSums(dfm_quin_gram_stem_nostop), decreasing =
TRUE)
quin_gram_s_ns <- data.frame(quin_gram_s_ns)
quin_gram_s_ns <- setDT(quin_gram_s_ns, keep.rownames = TRUE)
quin_gram_s_ns <- separate(quin_gram_s_ns, rn, c("word1", "word2", "word3",
"word4", "word5"), sep = " ")
write_csv(quin_gram_s_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quin_gram_s_ns.csv")
rm(dfm_quin_gram_stem_nostop)
rm(quin_gram_s_ns)
```

## neither stemming nor stop words

``` {r quingrams with neither stemming nor stop words, eval = TRUE}
dfm_quin_gram_nostem_nostop <- dfm(quin_gram, tolower = TRUE, stem = FALSE)
quin_gram_ns_ns <- sort(colSums(dfm_quin_gram_nostem_nostop), decreasing =
TRUE)
quin_gram_ns_ns <- data.frame(quin_gram_ns_ns)
quin_gram_ns_ns <- setDT(quin_gram_ns_ns, keep.rownames = TRUE)
~~~

~~~~
quin_gram_ns_ns <- separate(quin_gram_ns_ns, rn, c("word1", "word2",
"word3", "word4", "word5"), sep = " ")
write_csv(quin_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20sample/quin_gram_ns_ns.csv")
rm(dfm_quin_gram_nostem_nostop)
rm(quin_gram_ns_ns)
rm(quin_gram)
```

## And now for the test corpus left over for Quingrams

```{r test quin grams, eval = TRUE}
quin_gram_test <- tokens(testCorpus, remove_numbers = TRUE, remove_punct =
TRUE, remove_symbols = TRUE, what = "word", split_hyphens = TRUE,
remove_url = TRUE, ngrams = 5L,concatenator = " ")
dfm_quin_gram_nostem_nostop <- dfm(quin_gram_test, tolower = TRUE, stem =
FALSE)
quin_gram_ns_ns <- sort(colSums(dfm_quin_gram_nostem_nostop), decreasing =
TRUE)
quin_gram_ns_ns <- data.frame(quin_gram_ns_ns)
quin_gram_ns_ns <- setDT(quin_gram_ns_ns, keep.rownames = TRUE)
quin_gram_ns_ns <- separate(quin_gram_ns_ns, rn, c("word1", "word2",
"word3", "word4", "word5"), sep = " ")
write_csv(quin_gram_ns_ns, "/Users/mutecypher/Documents/Coursera/Capstone
Project/20test/quin_gram_test.csv")
rm(dfm_quin_gram_nostem_nostop)
rm(quin_gram_ns_ns)
rm(quin_gram_test)
rm(myCorpus)
rm(testCorpus)
```

## This is the end of the line
~~~~