New_First_Analysis

Michael Pearson

12/20/2017

Keep the n gram files - hadn't done that before.

```
knitr::opts_chunk$set(echo = TRUE)
library(tidytext, quietly = TRUE)
library(dplyr, quietly = TRUE)
library(readr, quietly = TRUE)
library(R.utils, quietly = TRUE)
library(tm, quietly = TRUE)
library(SnowballC, quietly = TRUE)
library(tidyr, quietly = TRUE)
library(data.table, quietly = TRUE)
eng_news <- read_file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/
eng_blogs <- read_file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project
eng_twitter <- read_file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Proje
blog_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/files/
tweet_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/files
news_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/files/
newslines <- countLines("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Proje
bloglines <- countLines("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Proje
tweetlines <- countLines("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Proje
blog25 <- readLines(blog_us, 25)
close(blog_us)
tweet25 <- readLines(tweet_us, 25)
close(tweet_us)
news25 <- readLines(news_us, 25)
close(news_us)
```

Exploring the data

Some basic examination of the three text sources: the News, the Blogs, the Tweets.

The news file has 205,243,643 characters, and 1,010,242 lines of text. The news file is 196.2775 MegaBytes.

The blog file has 208,623,081 characters, and 899,288 of text. The blog file is 200.4242 MegaBytes

The twitter file has 11,790,868 characters and 2,360,148 of text. The twitter file is 159.3641 MegaBytes

Beginning of text processing

```
newzchar <- nchar(news25)
sumnew <- sum(newzchar)
tweetchar <- nchar(tweet25)
sumtweet <- sum(tweetchar)
blogchar <- nchar(blog25)
sumblog <- sum(blogchar)
```

Begin the exploration by loading the full texts of the Swiftkey files…

```
tweet_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/file
tweet_all <- readLines(tweet_us, n= tweetlines, warn = FALSE, encoding = "UTF=8", skipNul = TRUE)
close(tweet_us)
love_it <- length(grep("love", tweet_all))
hate_it <- length(grep("hate", tweet_all))
phrase_it <- length(grep("A computer once beat me at chess, but it was no match for me at kickboxing",
blog_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/files
blog_all <- readLines(blog_us, n= bloglines, warn = FALSE, encoding = "UTF=8", skipNul = TRUE)
close(blog_us)
news_us <- file("/Users/mutecypher/Documents/Documents - Michael’s iMac/Coursera/Capstone Project/files
news_all <- readLines(news_us, n = newslines, warn = FALSE, encoding = "UTF=8", skipNul = TRUE)
romance <- length(grep("romantic date", news_all))
close(news_us)
long_tweet <- max(nchar(tweet_all[1:tweetlines]))
long_news <- max(nchar(news_all[1:newslines]))
long_blog <- max(nchar(blog_all[1:bloglines]))
```

More exploration of the texts

Some of these are from the quiz for the first week of the Capstone Project…

The phrase “A computer once beat me at chess, but it was no match for me at kickboxing” occurs 3 times in the twitter sample.

The word ‘love’ occurs 90,956 times in the twitter sample.

The word ‘hate’ occurs 22,138 times in the twitter sample.

The longest line in the twitter sample is 140 characters. Duh!

The longest line in the blog sample is 40,833 characters.

The longest line in the news sample is 11,384 characters.

The phrase “romantic date” occurs 5 times.

Getting rid of the profanity

I got a list of profanity from Google: full-list-of-bad-words-banned-by-google-txt-file_2013_11_26_04_53_31_867.txt

I will use this file to filter profanity from my sample of the corpus.

Now let’s create a sample of 50% of the text, load it into a corpus.

Then we will remove profanity (the ‘badwords’ file from Google), tidy the corpus using the ‘tidytext’ package - which follows tidy data procedures and makes one variable per column.

I use the file “full-list-of-bad-words-banned-by-google-txt-file_2013_11_26_04_53_31_867.txt” to create a list of profanity to remove from the samples.

We will remove non alphabetic characters, remove blanks, and then count word frequencies, and create tidy data frames for bigrams and trigrams.

```r
badwords <- readLines("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project
set.seed(1151960)
samp_per <- 0.2
sam_twit <- tweet_all[sample(1:length(tweet_all),samp_per*length(tweet_all))]
write_lines(sam_twit, "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project
sam_news <- news_all[sample(1:length(news_all),samp_per*length(news_all))]
write_lines(sam_news, "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project
sam_blog <- blog_all[sample(1:length(blog_all),samp_per*length(blog_all))]
write_lines(sam_blog, "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone Project
sam_Corpus <- VCorpus(DirSource("/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capsto
sam_tidy <- tidy(sam_Corpus)
data("stop_words")
```

Two versions of the corpus are loaded

```r
tidy_sentences <- data.table(sam_tidy) %>% unnest_tokens(sentences, text, token = "sentences")
text_tokens <- data.table(sam_tidy) %>% unnest_tokens(word, text, token = "words")
text_tokens$word <- gsub("[^[:alpha:] | ^[:punct:]]" , " ", text_tokens$word)
text_tokens$word   <- gsub("-", " ", text_tokens$word)
tidy_sentences$sentences <- gsub("[^[:alpha:] | ^[:punct:]]", " ", tidy_sentences$sentences)
tidy_sentences$sentences <- gsub("-", " ", tidy_sentences$sentences)
tidy_sentences$sentences <- gsub("  ", " ", tidy_sentences$sentences)
```

```r
text_bigrams <- sam_tidy %>% unnest_tokens(bigram, text, token = "ngrams", n=2)
text_bigrams <- text_bigrams$bigram
write.csv(text_bigrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
rm(text_bigrams)
sen_bigrams <- tidy_sentences %>% unnest_tokens(bigrams, sentences, token = "ngrams", n = 2)
sen_bigrams <- sen_bigrams$bigrams
write.csv(sen_bigrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
rm(sen_bigrams)
```

Now the trigrams

```r
text_trigrams <- sam_tidy %>% unnest_tokens(trigram, text, token = "ngrams", n=3)
text_trigrams <- text_trigrams$trigram
write.csv(text_trigrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capston
rm(text_trigrams)
sen_trigrams <- tidy_sentences %>% unnest_tokens(trigrams, sentences, token = "ngrams", n = 3)
sen_trigrams <- sen_trigrams$trigrams
write.csv(sen_trigrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
rm(sen_trigrams)
```

and the quadgrams

```r
text_quadgrams <- sam_tidy %>% unnest_tokens(quadgram, text, token = "ngrams", n=4)
text_quadgrams <- text_quadgrams$quadgram
write.csv(text_quadgrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capsto
rm(text_quadgrams)
sen_quadgrams <- tidy_sentences %>% unnest_tokens(quadgrams, sentences, token = "ngrams", n = 4)
sen_quadgrams <- sen_quadgrams$quadgrams
```

```
write.csv(sen_quadgrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
rm(sen_quadgrams)
```

and the quingrams

```
text_quingrams <- sam_tidy %>% unnest_tokens(quingram, text, token = "ngrams", n=5)
text_quingrams <- text_quingrams$quingram
write.csv(text_quingrams ,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capsto
rm(text_quingrams)
sen_quingrams <- tidy_sentences %>% unnest_tokens(quingrams, sentences, token = "ngrams", n = 5)
sen_quingrams <- sen_quingrams$quingrams
write.csv(sen_quingrams,file = "/Users/mutecypher/Documents/Documents - Michael's iMac/Coursera/Capstone
rm(sen_quingrams)
```

and let's see how that goes