SwiftKey Corpus Exploratory Analysis

Michael Nichols

May 10, 2018

Exploratory Data Analysis of the SwiftKey Corpus

Data Science Specialization Capstone Project

Tasks to accomplish

1. Exploratory analysis - perform a thorough exploratory analysis of the data, understanding the distribution of words and relationship between the words in the corpora.
2. Understand frequencies of words and word pairs - build figures and tables to understand variation in the frequencies of words and word pairs in the data.

Initial Exploratory Analysis

Descriptive Statistics of the Full Datasets: Blogs, News, Twitter

file

totalLines

totalWords

totalChars

averageWords

minWords

maxWords

averageChars

minChars

maxChars

blogs

44965

1454377

9425914

32.34

1

1030

209.63

5

7293

news

3863

105488

719493

27.31

1

196

186.25

9

1220

twitter

118008

1277890

7954230

10.83

1

43

67.40

6

149

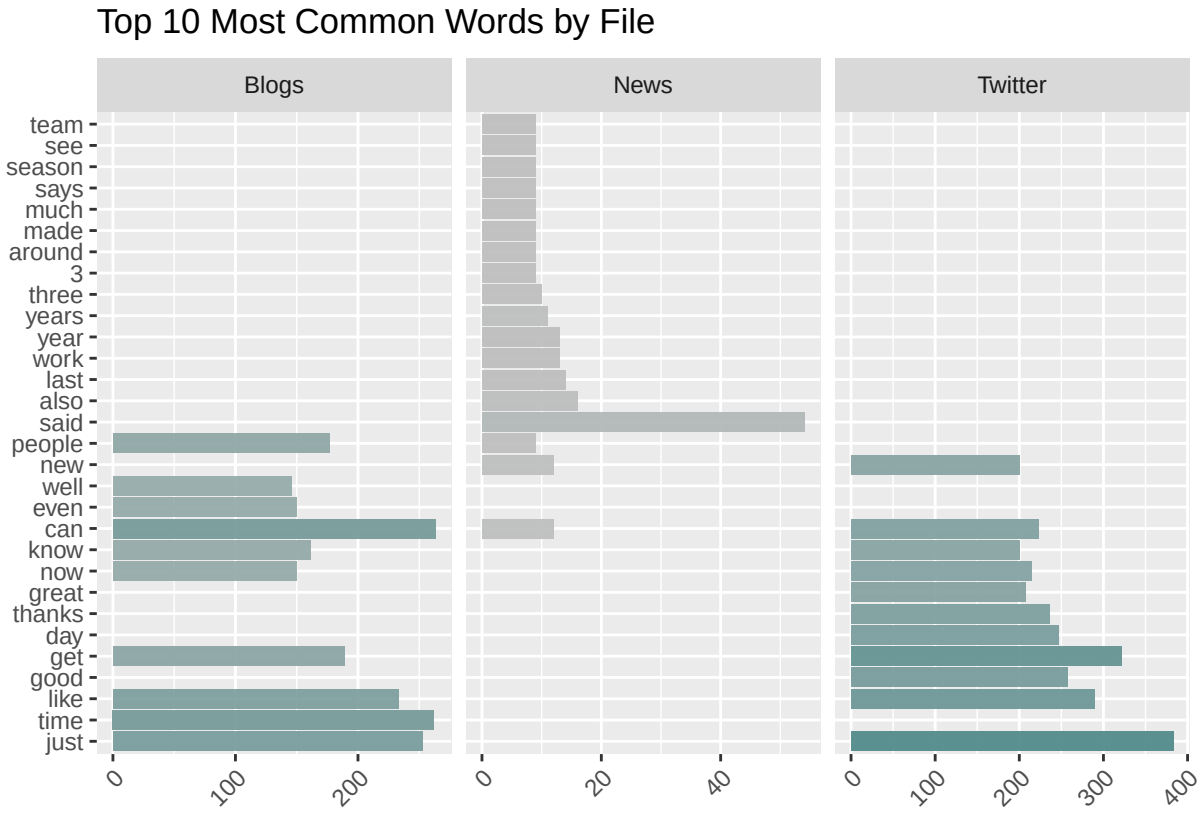
Create tokenized version of each dataset

```
#tokenize the data frames (split each word from each document into its own row)
#also remove 'stopwords': extremely common words not valuable for an analysis (e.g. as, of, the , a, ..
tokenized_blogs <- clean_blogs_df %>%
        unnest_tokens(output = word, input = text) %>%
        anti_join(get_stopwords())

tokenized_news <- clean_news_df %>%
        unnest_tokens(output = word, input = text) %>%
        anti_join(get_stopwords())

tokenized_twitter <- clean_twitter_df %>%
        unnest_tokens(output = word, input = text) %>%
        anti_join(get_stopwords())
```

Further Exploratory Visualizations

Plot of top 10 most common words across each file

```
## Warning: package 'bindrcpp' was built under R version 3.4.3
```



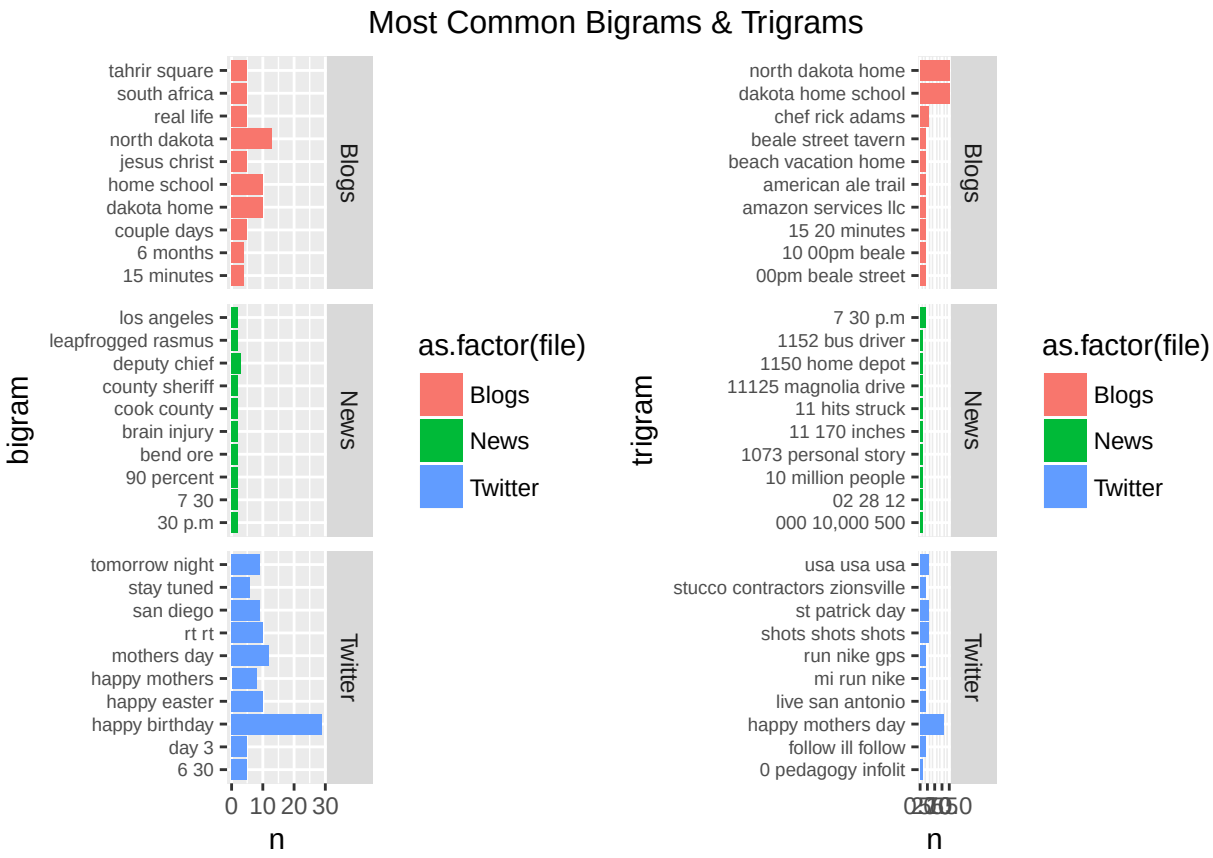
N-gram Analysis

Per file, the following code gathers the top 10 most common 2-word (bigram) and 3-word (trigram) strings. This code ignore stop words (e.g. as, of, the , a, . . . , etc.) to show drive more substantial insights.

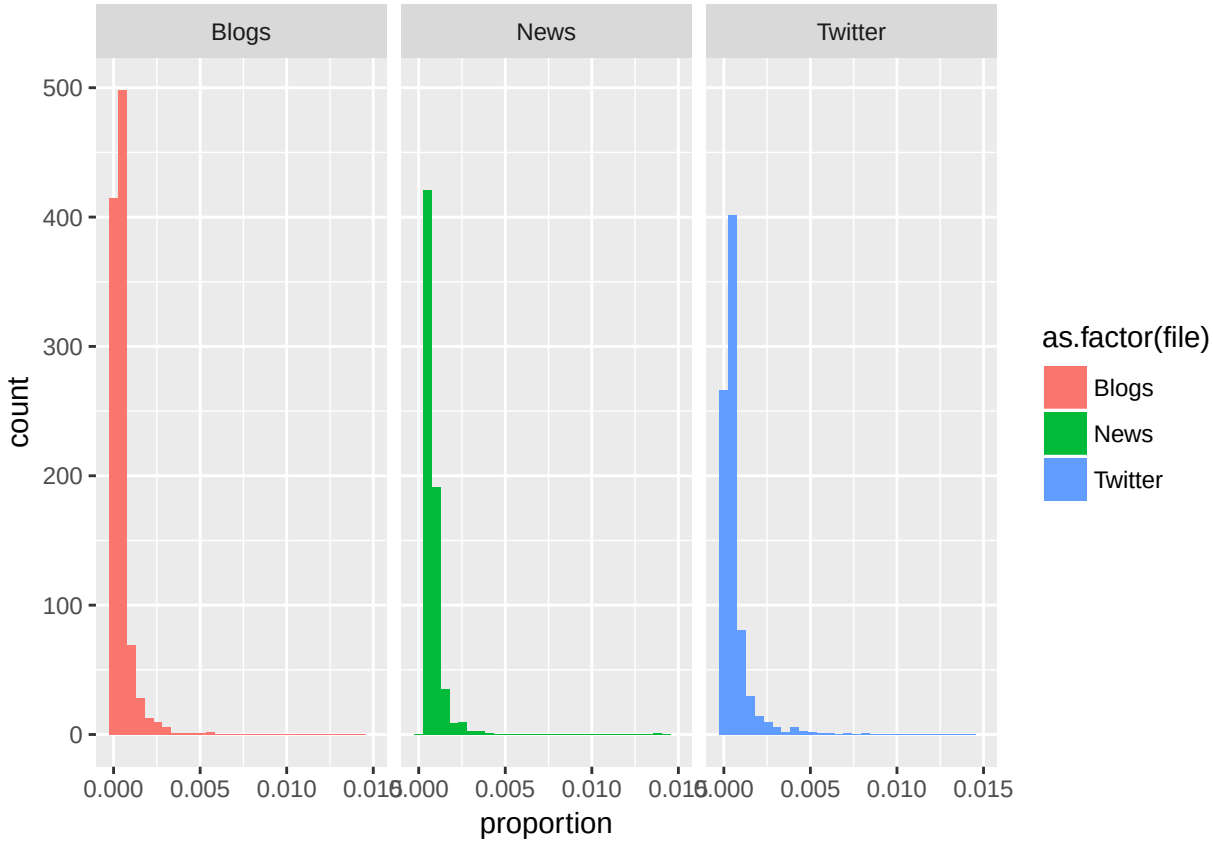
```
#n-gram analysis
#bigrams
bigram_Analysis <- function(text, filetype) {
        text %>%
                unnest_tokens(bigram, text, token = "ngrams", n = 2) %>%
                tidyr::separate(bigram, c("word1", "word2"), sep = " ") %>%
                na.omit() %>%
                filter(!word1 %in% stop_words$word,
                       !word2 %in% stop_words$word) %>%
                count(word1, word2, sort = TRUE) %>%
                top_n(n = 10, wt = n) %>% #selects only top 10 values
                slice(row_number(1:10)) %>% #prevents ties (i.e. several bigrams with the 10th highest
                mutate(bigram = paste(word1, word2, sep = " ")) %>%
                mutate(file = filetype)
}

blogs_bigram <- bigram_Analysis(clean_blogs_df, "Blogs")
news_bigram <- bigram_Analysis(clean_news_df, "News")
twitter_bigram <- bigram_Analysis(clean_twitter_df, "Twitter")
```

```r
#combine for easy visualizations
bigram_all <- as.data.frame(rbind.data.frame(blogs_bigram, news_bigram, twitter_bigram))

#trigrams
trigram_Analysis <- function(text, filetype) {
        text %>%
                unnest_tokens(trigram, text, token = "ngrams", n = 3) %>%
                tidyr::separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%
                na.omit() %>%
                filter(!word1 %in% stop_words$word,
                       !word2 %in% stop_words$word,
                       !word3 %in% stop_words$word) %>%
                count(word1, word2, word3, sort = TRUE) %>%
                top_n(n = 10, wt = n) %>%
                slice(row_number(1:10)) %>%
                mutate(trigram = paste(word1, word2, word3, sep = " ")) %>%
                mutate(file = filetype)
}

blogs_trigram <- trigram_Analysis(clean_blogs_df, "Blogs")
news_trigram <- trigram_Analysis(clean_news_df, "News")
twitter_trigram <- trigram_Analysis(clean_twitter_df, "Twitter")

#combine for easy visualizations
trigram_all <- as.data.frame(rbind.data.frame(blogs_trigram, news_trigram, twitter_trigram))
```

Visualize the Bigrams & Trigrams



Plot of concentration of certain words (only shows terms that make up .50 of the total) Plots with higher bars on the left have more unique terms (more instances where the frequency of a certain word is very small). *Below the plot is a table displaying the number of unique words required to hit the specified percentages of all words in the text* Hypothetical example to illustrate this concept: The word 'the' could occur 200 times in a 10,000 word document, thus consisting of 2% of all words in a document.



Unique Terms to Cover X Percent of Total Words

file

0.1

0.3

0.5

0.7

0.9

1

Blogs

38

302

1076

3220

10314

14945

News

43

269

693

1462

2230

2596

Twitter

24

191

855

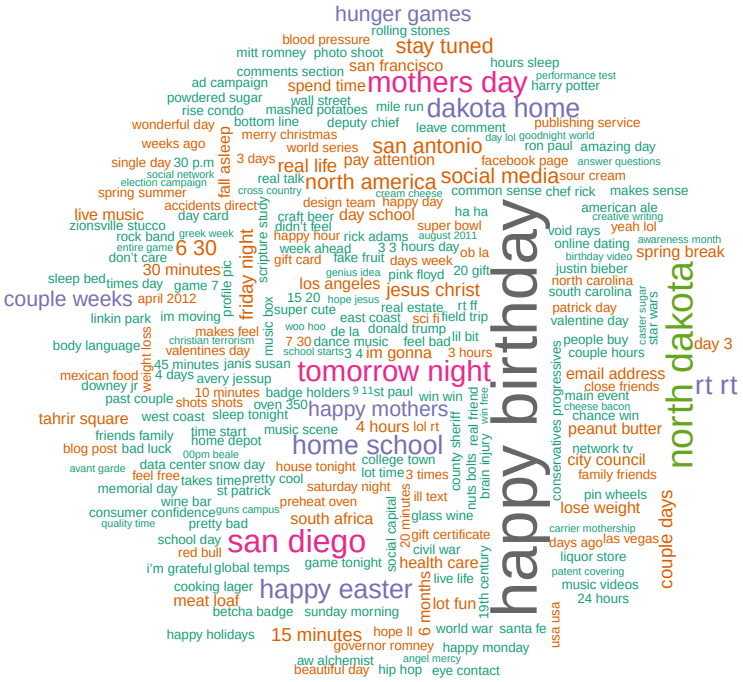
3501

12158

16571

Appendix

Wordclouds Now looking at all the files together without first filtering for the most common words, create 3 wordcloud: individual word, bigrams, and trigrams. – This visualization could be well suited for a final data product (Shiny app).


happy birthday
hunger games
rolling stones
blood pressure
stay tuned
mitt romney
photo shoot
hours sleep
san francisco
performance test
comments section
harry potter
ad campaign
spend time
mothers day
powdered sugar
wall street
rise condo
mashed potatoes
mile run
dakota home
publishing service
wonderful day
bottom line
deputy chief
leave comment
day lol goodnight world
merry christmas
weeks ago
world series
san antonio
ron paul
amazing day
single day 30 p.m
3 days
real life
pay attention
facebook page
answer questions
social network
election campaign
real talk
north america
social media
sour cream
spring summer
cross country
cream cheese
common sense
chef rick
makes sense
accidents direct
design team
happy day
live music
day card
craft beer
day school
ha ha
american ale
creative writing
zionsville stucco
didn't feel
super bowl
void rays
yeah lol
rock band
greek week
happy hour
rick adams
august 2011
online dating
awareness month
entire game
6 30
week ahead
3 3 hours day
birthday video
spring break
don't care
gift card
take fruit
days week
ob la
30 minutes
genius idea
pink floyd
20 gift
justin bieber
sleep bed
times day
game 7
los angeles
jesus christ
north carolina
couple weeks
april 2012
15 20
hope jesus
real estate
rt ff
south carolina
patrick day
linkin park
im moving
super cute
east coast
sci fi
field trip
valentine day
makes feel
woo hoo
de la
donald trump
body language
christian terrorism
7 30
dance music
feel bad
lil bit
people buy
day 3
valentines day
school starts
3 4 im gonna
3 hours
couple hours
45 minutes
janis susan
tomorrow night
email address
mexican food
4 days
avery jessup
close friends
rt rt
downey jr
10 minutes
badge holders
9 11
st paul
win win
main event
past couple
shots
shots
oven 350
happy mothers
cheese bacon
tahrir square
west coast
sleep tonight
chance win
4 hours
lol rt
peanut butter
friends family
time start
music scene
home depot
home school
network tv
blog post
bad luck
00pm beale
college town
city council
avant garde
data center
snow day
house tonight
lot time
3 times
family friends
feel free
takes time
pretty cool
memorial day
st patrick
saturday night
pin wheels
wine bar
preheat oven
ill text
lose weight
consumer confidence
guns campus
glass wine
quality time
pretty bad
south africa
carrier mothership
school day
san diego
gift certificate
days ago
las vegas
red bull
civil war
liquor store
i'm grateful
global temps
game tonight
health care
patent covering
cooking lager
happy easter
live life
music videos
meat loaf
lot fun
24 hours
betcha badge
sunday morning
happy holidays
15 minutes
hope ll
6 world war
santa fe
governor romney
happy monday
aw alchemist
angel mercy
beautiful day
hip hop
eye contact
north dakota
rt rt
day 3
couple days
fall asleep
friday night
scripture study
profile pic
weight loss
music box
conservatives progressives
caster sugar
star wars
real friend
win free
county sheriff
brain injury
nuts bolts
social capital
20 minutes
6 months
19th century
usa usa

happy mothers day

dakota home school

north dakota home