Statistical Language Models
2019
Week 3 part 1

Dr. Dave Campbell
davecampbell@math.carleton.ca

Approximate Course Outline

Week 1: ShinyApps and Dashboarding

Week 2: TidyText & obtaining data, dealing with time events

Week 3: Regular Expressions; Word co-occurrence explorations

Week 4: Sentiment Analysis; Stochastic process models

Week 5: Exponential models for time between events.

Week 6: Bayesian Basics; Author attribution models; hierarchical models

Week 7: MCMC Diagnostics

Week 8: Embeddings and Word2Vec; Cryptography

Week 9: Clustering; Latent Dirichlet Allocation and topic models.

Week 10: Variational Inference

Week 11: Getting Fancier with Language Models

Week 12: Student projects and presentations

https://genius.com song lyrics:

```
library(genius)

library(rvest)

library(tidyverse)
```

Lyrics as a nibble

```
Artist = "Ed Sheeran"

Song = "I See Fire"

Lyrics = genius_lyrics(artist=Artist ,song=Song)
```

Find all tracks from an album

```
Artist = "Pink Floyd"

Album = "The Dark Side Of The Moon"

tracklist = genius_tracklist(artist=Artist, album = Album)

#All lyrics from an album:

Lyrics = NULL

for(songNumber in dim(tracklist)[1]:1){

Lyrics = rbind(Lyrics,genius_lyrics(artist=Artist ,song= tracklist$track_title[songNumber]))

}
```

```
LyricsTib =  Lyrics %>% unnest_tokens(output = word,input = lyric, token
= "words")
```

```
library(rtweet)

library(ROAuth)

See the tweet vignette on authentication
```

Vignettes from package 'rtweet'

rtweet::FAQ	FAQ
rtweet::auth	Obtaining and using access tokens
rtweet::intro	Intro to rtweet
rtweet::stream	Live streaming tweets

```
library(rtweet)

library(tidytext);

library(tidyverse)

DS = search_tweets("#datascience",n = 10000)

DV = search_tweets("#Davos OR #Davos20",n=10000,retryonratelimit =
TRUE)
```

Tweet timing

```
Tweets2use = DS

weekdays(Tweets2use$created_at)

table(weekdays(Tweets2use$created_at))

barplot(table(weekdays(Tweets2use$created_at)),las=2)

barplot(table(weekdays(Tweets2use$created_at))
[c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")],
las=2)

plot(table(weekdays(Tweets2use$created_at)),las=2)
```

```
Tweets2use = DV

WordsCounted = Tweets2use %>%

  unnest_tokens(      output = word,input = text, token = "words")  %>%

    count(word,sort = TRUE)

barplot(height=WordsCounted$n[1:50],names.arg =WordsCounted$word[1:50], las=2)
```

Wordcloud

```
# produce a word cloud


library(wordcloud)


wordcloud(WordsCounted$word,   WordsCounted$n,
min.freq=500, colors=rainbow(8))
```

There are often a lot of useless words. We may want to remove them to do anything meaningful.

```
TweetTibble = DV

FewerWordsCounted = TweetTibble %>%

  unnest_tokens(      output = word,input = text, token = "words")  %>%

    anti_join(stop_words) %>%

  count(word,sort = TRUE)
```

We can make a wordcloud of the occurences

```
library(wordcloud)

Hashtag = "#Davos"

text(x=0.5, y=0.5, paste("Commonly used words
from",Hashtag,"Tweets" ))

wordcloud(FewerWordsCounted$word,FewerWordsCounted$n,

          colors = rainbow(8), min.freq = 500)
```

Time

```
sort(Tweets2use$created_at)

plot(sort(Tweets2use$created_at))

max(Tweets2use$created_at)-min(Tweets2use$created_at)

write.csv(sort(Tweets2use$created_at),file="OUTATIME.csv")
```

```
MoreTime = read.csv(file="OUTATIME.csv")

head(MoreTime)

plot(MoreTime[,2])

max(MoreTime[,2])-min(MoreTime[,2])
```

class(DS$created_at)

class(MoreTime[,2])

Portable Operating System Interface: POSIX

Time contains (hidden) integer time from (default) origin of 1970-01-01 00:00.00 UTC

UTC = Universal Time Coordinated aka Greenwhich Mean Time (from 0^{o} longitude)

EST = GMT - 5

```
PTime = as.POSIXct(MoreTime[,2])

plot(PTime)

max(PTime)-min(PTime)
```

```
Time2use = Tweets2use$created_at


trunc(MoreTime[,2],"hours")#fails

trunc(as.POSIXct(MoreTime[,2]),"hours")

trunc(Time2use,"days")

cut(Time2use,"years")
```

Extracting time

strftime(MoreTime[,2], format="%H:%M:%S")

#Extract minute of the hour (in seconds):

```
difftime( Time2use,
        trunc(Time2use,"hours"))
```

```
hist(difftime( Time2use,  trunc(Time2use,"hours")))
```

```
sentence = "RT: @JulioTrujillo_: Here is the amazing evolution @Evolution of #IofT #Analytics via @markm https://t.co/dTrSNy7vHF"

# First we will remove retweet entities from tweets

sentence2 = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", " ", sentence)

# Then remove all @someone

sentence3 = gsub("@\\w+", " ", sentence2)

# remove all the punctuation except apostrophe

sentence4 = gsub("(?!')[[:punct:]]", "", sentence3,    perl = T)

# remove all the control chracters, like \n or \r

 sentence5 = gsub("[[:cntrl:]]", "", sentence4)

# remove numbers, we need only text for analytics

sentence6 = gsub("[[:digit:]]", "", sentence5)

# unify encoding to avoid tolower() error caused by
# emoji
sentence7 = iconv(sentence6, "ASCII", "UTF-8", sub = "")

# convert to lower case

sentence8 = tolower(sentence7) # remove url links

sentence9 = gsub("http\\w+", "", sentence8)

# remove unnecessary spaces (white spaces, tabs # etc)
sentence10 = gsub("[ \t]{2,}", " ", sentence9)

sentence11 = gsub("^\\s+|\\s+$", "", sentence10)
```