Statistical Language Models 2019 Week 4 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://www.tidytextmining.com/twitter.html

Great book.

Available irl and online

HTML

```
<p>Directly parsing Canada's <a href="https://
en.wikipedia.org/wiki/
List_of_National_Parks_of_Canada"
title="National Parks of Canada">National
Parks</a> table should be much easier than html
into regular expressions.</p>
```

WebScraping

```
library(rvest)

file = read_html("https://en.wikipedia.org/wiki/
List_of_National_Parks_of_Canada")

out = html_table(html_nodes(file, "table")[[1]])



dim(out)

out[,1]

out[,2:3]
```

fixing size, km^2:

```
head(out)

out[,5]

step1 =  gsub(x=out[,5],pattern =     ,replacement =   )

step2 =    gsub(x=step1,     pattern =      ,replacement=   )

km2 = as.numeric(step2)
```

fixing size, sq.mi.:

```
 head(out)

out[,5]

step1 =  gsub(x=out[,5],pattern =     ,replacement =   )

step2 =    gsub(x=step1,    pattern =      ,replacement=   )

step3 =    gsub(x=step2,    pattern =      ,replacement=   )

sqmi = as.numeric(step3)
```

Touch ups

fixing location :

```
out[,3]

prov =    gsub(out[,3],pattern="   ",replacement="  ")
```

fixing year:

```
out[,2]

year = as.numeric(gsub(out[,4],pattern="  ",replacement="  "))
```

Filling in the table

```
NatParks = data.frame(name=out[,1],
          year=year,
          size=km2,
          location = prov)
```

```
#load our libraries

library(stringr)

library(tidyr)

library(tidyverse)

 library(tidytext)
```

SENTIMENT LEXICONS

```
library(textdata)
```

Sentiment Analysis is all about comparing words in a document to a lexicons:

AFINN from Finn Årup Nielsen, AFINN {-5,-4,…,5} rating its severity of positive or negative sentiment.

bing from Bing Liu and collaborators {-1, 1} binary positive or negative, neutral words are not in the list

nrc from Saif Mohammad and Peter Turney {"trust","fear","negative", "sadness" ,"anger","surprise","positive" , "disgust","joy", "anticipation"} nrc puts each word into a sentiment category.

```
get_sentiments("afinn")
```

```
library(gutenbergr)

JaneBooks = gutenberg_works(author == "Austen, Jane") %>%

      gutenberg_download(meta_fields = "title")

# meta fields lets us keep track of additional information, here I'm keeping the book title name



  JaneBooks = gutenberg_download(gutenberg_works(author == "Austen, Jane"))

tidy_books = JaneBooks %>%

       group_by(title) %>%      #  group by

         mutate(linenumber = row_number(),

              chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]", ignore_case = TRUE)))) %>%

             ungroup() %>%

               unnest_tokens(word, text)
```

Tracking Joy in Emma (by filtering)

```
nrc_joy = get_sentiments("nrc") %>%   filter(sentiment == "joy")



tidy_books %>%

  filter(title == "Emma") %>%

  inner_join(nrc_joy) %>%

group_by(chapter)%>%

  count(word, sort = TRUE) # could also count within chapter
```

BING LEXICON

```
jane_austen_sentiment = tidy_books %>%

  inner_join(get_sentiments("bing")) %>%

#  count(title, index = linenumber %/% 80, sentiment) %>% # look within
blocks of 80 lines

 count(title, index = chapter, sentiment) %>% # look within chapters

  spread(sentiment, n, fill = 0) %>%

  mutate(sentiment = positive - negative)
```

Plot sentiment across the book

```
library(ggplot2)

ggplot(jane_austen_sentiment,     # data to use

       aes(index, sentiment, fill = title)) +   # variables: index[line number],
sentiment, and colour each book individually

  geom_col(show.legend = FALSE) +           # style

  facet_wrap(~title, ncol = 2, scales = "free_x")  # colouring
```

AFINN Lexicon

```
jane_austen_afinn = tidy_books %>%
  inner_join(get_sentiments("afinn")) %>%
#  group_by(title, index = linenumber %/% 80) %>% #group by discrete chins of lines
  group_by(title, index = chapter) %>%  # group by chapters
  summarise(sentiment = sum(value)) %>%      #add up sentiment within a chapter
  mutate(method = "AFINN")



ggplot(jane_austen_afinn,     # data to use
       aes(index, sentiment, fill = title)) +   # variables: index[line number], sentiment, and colour each book individually
  geom_col(show.legend = FALSE) +            # style
  facet_wrap(~title, ncol = 2, scales = "free_x")  # split by title
```

NRC

```
jane_austen_nrc = tidy_books %>%
  inner_join(get_sentiments("nrc")) %>%
    filter(sentiment %in% c("positive", "negative"))%>%
                mutate(method = "NRC") %>%
  count(title, index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)



ggplot(jane_austen_nrc,     # data to use
     aes(index, sentiment, fill = title)) +   # variables: index[line number], sentiment, and colour each book individually
  geom_col(show.legend = FALSE) +            # style
  facet_wrap(~title, ncol = 2, scales = "free_x")
```

More NRC sentiments

```
jane_austen_nrc_full = tidy_books %>%
  inner_join(get_sentiments("nrc")) %>%
  count(title, index =chapter, sentiment)%>%
   mutate(method = "NRC")
ggplot(jane_austen_nrc_full, aes(index, n,color = sentiment)) +
    geom_line() +
  facet_grid(~ sentiment)+
  facet_wrap(~title, ncol = 2, scales = "free_x")  # split by title
```