Assignment 12

Allison Tessman

2024-04-04

Question 1. Working with a text dataset containing Amazon reviews for an iPhone model, do the following:

* Create a list of pairs of words that may contain stop words and plot the most frequent pairs.

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2)  
library(tidytext)  
  
df <- read\_csv('https://bryantstats.github.io/math475/assignments/Amazon\_Unlocked\_Mobile2.csv')

## Rows: 1451 Columns: 2  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Product\_Name, texts  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

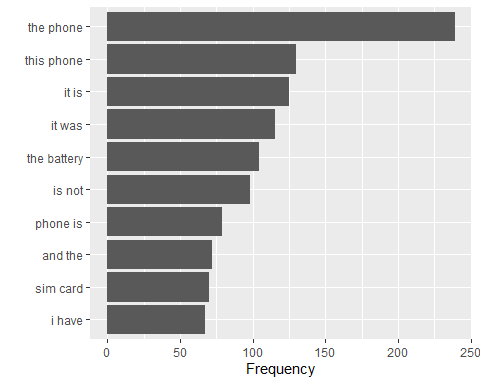
df %>%   
 group\_by(Product\_Name) %>%   
 count(sort = TRUE)

## # A tibble: 1 × 2  
## # Groups: Product\_Name [1]  
## Product\_Name n  
## <chr> <int>  
## 1 Apple iPhone 4s 8GB Unlocked Smartphone w/ 8MP Camera, White (Certified… 1451

#Create list of pairs of words  
df\_bigrams <- df %>%  
 unnest\_tokens(input = texts, output = bigram, token = "ngrams", n = 2) %>%   
 filter(!is.na(bigram))  
  
df\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 10,117 × 2  
## bigram n  
## <chr> <int>  
## 1 the phone 239  
## 2 this phone 130  
## 3 it is 125  
## 4 it was 115  
## 5 the battery 104  
## 6 is not 98  
## 7 phone is 79  
## 8 and the 72  
## 9 sim card 70  
## 10 i have 67  
## # ℹ 10,107 more rows

#Top most frequent pairs  
df\_bigrams %>%  
 count(bigram, sort = TRUE) %>%   
 head(10) %>%   
 ggplot(aes(x = n, y = reorder(bigram, n))) +  
 geom\_col() +  
 labs(y = '', x = 'Frequency')

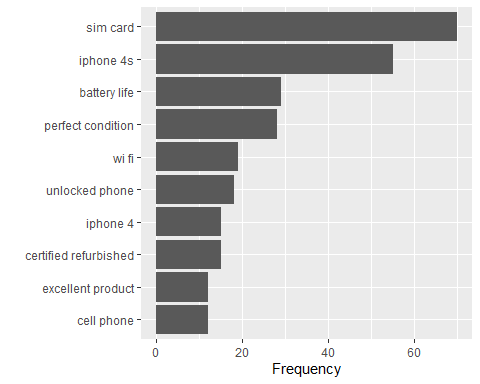


* Create a list of pairs of words that do not contain stopwords and plot the most frequent pairs.

#Remove stop words  
bigrams\_separated <- df\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_filtered <- bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
#New bigram counts:  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
bigram\_counts

## # A tibble: 1,425 × 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 sim card 70  
## 2 iphone 4s 55  
## 3 battery life 29  
## 4 perfect condition 28  
## 5 wi fi 19  
## 6 unlocked phone 18  
## 7 certified refurbished 15  
## 8 iphone 4 15  
## 9 cell phone 12  
## 10 excellent product 12  
## # ℹ 1,415 more rows

#Top most frequent pairs  
bigram\_counts$pairs = paste0(bigram\_counts$word1, " ", bigram\_counts$word2)  
  
bigram\_counts %>%   
 head(10) %>%   
 ggplot(aes(x = n, y = reorder(pairs, n))) +  
 geom\_col() +  
 labs(y = '', x = 'Frequency')



* Visualize the pairs of words (bigrams)

library(igraph)

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:lubridate':  
##   
## %--%, union

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## The following objects are masked from 'package:purrr':  
##   
## compose, simplify

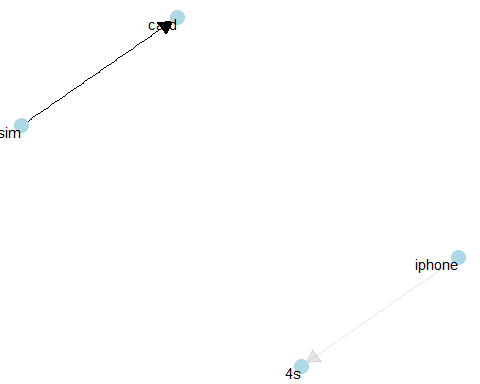
## The following object is masked from 'package:tidyr':  
##   
## crossing

## The following object is masked from 'package:tibble':  
##   
## as\_data\_frame

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

bigram\_graph <- bigram\_counts %>%  
 filter(n > 50) %>%  
 graph\_from\_data\_frame()  
  
library(ggraph)  
set.seed(2024)  
  
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()



* Plot the context sentiment analysis

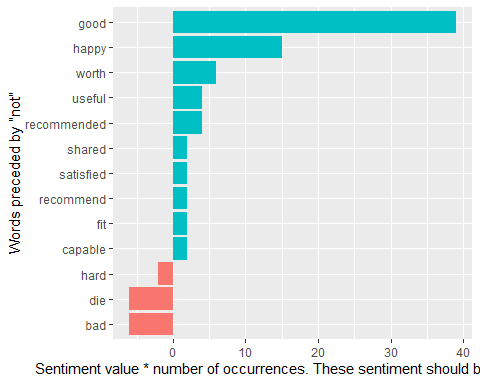
#Words preceded by not  
bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 137 × 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not unlocked 45  
## 2 not work 44  
## 3 not working 27  
## 4 not buy 19  
## 5 not be 18  
## 6 not a 14  
## 7 not sure 14  
## 8 not good 13  
## 9 not hold 13  
## 10 not even 12  
## # ℹ 127 more rows

AFINN <- get\_sentiments("afinn")  
  
not\_words <- bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word2, value, sort = TRUE)  
  
not\_words

## # A tibble: 13 × 3  
## word2 value n  
## <chr> <dbl> <int>  
## 1 good 3 13  
## 2 happy 3 5  
## 3 worth 2 3  
## 4 bad -3 2  
## 5 capable 1 2  
## 6 die -3 2  
## 7 fit 1 2  
## 8 hard -1 2  
## 9 recommended 2 2  
## 10 shared 1 2  
## 11 useful 2 2  
## 12 recommend 2 1  
## 13 satisfied 2 1

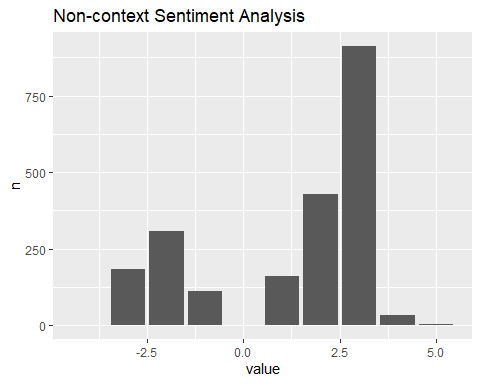
library(ggplot2)  
  
not\_words %>%  
 mutate(contribution = n \* value) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(n \* value, word2, fill = n \* value > 0)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = "Sentiment value \* number of occurrences. These sentiment should be reversed",  
 y = "Words preceded by \"not\"")



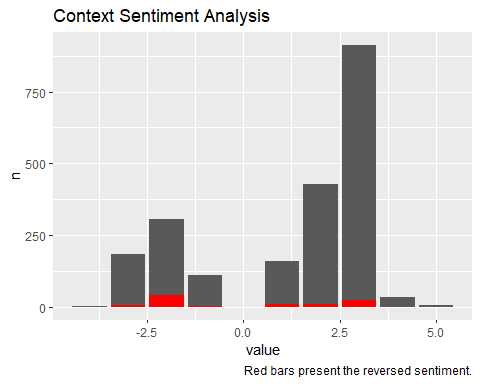
#Identify words that connect with not, no, never, without... then reverse the sentiment value of these words  
stop\_word2 = tibble(word = c(letters, LETTERS, "oh", 'just'))  
AFINN <- get\_sentiments("afinn")  
  
sentiment1 = df %>%  
 unnest\_tokens(input = texts, output = word) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(stop\_word2) %>%   
 inner\_join(get\_sentiments("afinn")) %>%   
 filter(!is.na(value)) %>%   
 count(value)

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`

sentiment1 %>%   
 ggplot(aes(x = value, y = n))+  
 geom\_col()+   
 labs(title = 'Non-context Sentiment Analysis')



negation\_words <- c("not", "no", "never", "without")  
  
sentiment2 = bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word1, word2, value, sort = TRUE) %>%   
 select(value, n) %>%   
 group\_by(value) %>%   
 summarise(n = sum(n)) %>%   
 arrange(value)  
  
  
ggplot()+  
 geom\_col(data = sentiment1, aes(x =value, y = n))+  
 geom\_col(data = sentiment2, aes(x =value, y = n), fill = 'red')+  
 labs(title = 'Context Sentiment Analysis',  
 caption = "Red bars present the reversed sentiment.")



Question 2. Do Question 1 on your own text data set.

library(tidyverse)  
library(ggplot2)  
library(tidytext)  
  
df <- read\_csv('~/Applied Analystics SAS Prog/mymath475/CNNtext.csv')

## Rows: 11490 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): id, article, highlights  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

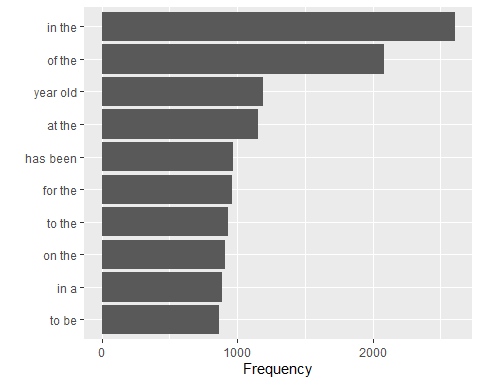
df %>%   
 group\_by(id) %>%   
 count(sort = TRUE)

## # A tibble: 11,490 × 2  
## # Groups: id [11,490]  
## id n  
## <chr> <int>  
## 1 000571afe702684d90c1d222ce70b1e1375c1016 1  
## 2 000642916e3a6c33411c617cf2f3c134a206fba8 1  
## 3 00110802bc6eae0e8e4d3d22e27f458f41be2b22 1  
## 4 00119229166ae09a6ef25c0e10b101ef9eb9cca3 1  
## 5 0013aa16650fbcfbe6edb16ac614ad174cb5d1cf 1  
## 6 00180b7ce54794a52766d795506a94071f7c055b 1  
## 7 001ebaa80dca4a65adf2178b132113cb9e3d5431 1  
## 8 00200e794fa41d3f7ce92cbf43e9fd4cd652bb09 1  
## 9 00217448b38d81a23db66ac362bee25056f58fab 1  
## 10 0021fe8d65bd0d6d76d5fefba2ac02f0c48a43f4 1  
## # ℹ 11,480 more rows

#Create list of pairs of words  
df\_bigrams <- df %>%  
 unnest\_tokens(input = highlights, output = bigram, token = "ngrams", n = 2) %>%   
 filter(!is.na(bigram))  
  
df\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 295,875 × 2  
## bigram n  
## <chr> <int>  
## 1 in the 2604  
## 2 of the 2083  
## 3 year old 1194  
## 4 at the 1153  
## 5 has been 969  
## 6 for the 965  
## 7 to the 932  
## 8 on the 907  
## 9 in a 888  
## 10 to be 864  
## # ℹ 295,865 more rows

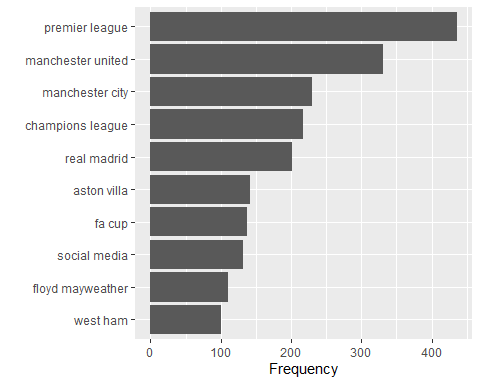
#Top most frequent pairs  
df\_bigrams %>%  
 count(bigram, sort = TRUE) %>%   
 head(10) %>%   
 ggplot(aes(x = n, y = reorder(bigram, n))) +  
 geom\_col() +  
 labs(y = '', x = 'Frequency')



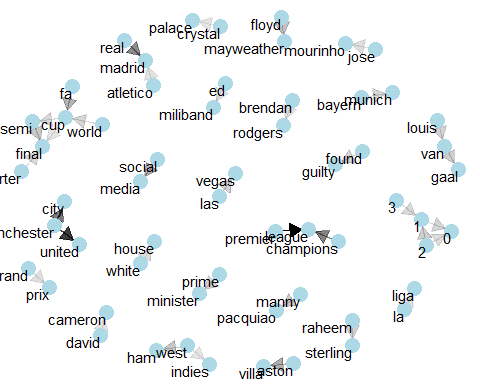
#Remove stop words  
bigrams\_separated <- df\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_filtered <- bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
#New bigram counts:  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
bigram\_counts

## # A tibble: 114,114 × 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 premier league 436  
## 2 manchester united 331  
## 3 manchester city 230  
## 4 champions league 217  
## 5 real madrid 201  
## 6 aston villa 142  
## 7 fa cup 137  
## 8 social media 132  
## 9 floyd mayweather 111  
## 10 west ham 101  
## # ℹ 114,104 more rows

#Top most frequent pairs  
bigram\_counts$pairs = paste0(bigram\_counts$word1, " ", bigram\_counts$word2)  
  
bigram\_counts %>%   
 head(10) %>%   
 ggplot(aes(x = n, y = reorder(pairs, n))) +  
 geom\_col() +  
 labs(y = '', x = 'Frequency')



library(igraph)  
bigram\_graph <- bigram\_counts %>%  
 filter(n > 50) %>%  
 graph\_from\_data\_frame()  
  
library(ggraph)  
set.seed(2024)  
  
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()



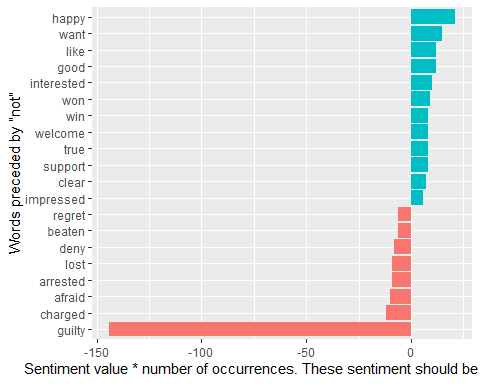
#Words preceded by not  
bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 582 × 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not be 87  
## 2 not to 72  
## 3 not been 59  
## 4 not guilty 48  
## 5 not have 42  
## 6 not a 32  
## 7 not the 24  
## 8 not know 22  
## 9 not yet 21  
## 10 not allowed 16  
## # ℹ 572 more rows

AFINN <- get\_sentiments("afinn")  
  
not\_words <- bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word2, value, sort = TRUE)  
  
not\_words

## # A tibble: 112 × 3  
## word2 value n  
## <chr> <dbl> <int>  
## 1 guilty -3 48  
## 2 want 1 15  
## 3 clear 1 7  
## 4 happy 3 7  
## 5 like 2 6  
## 6 afraid -2 5  
## 7 agree 1 5  
## 8 interested 2 5  
## 9 leave -1 5  
## 10 charged -3 4  
## # ℹ 102 more rows

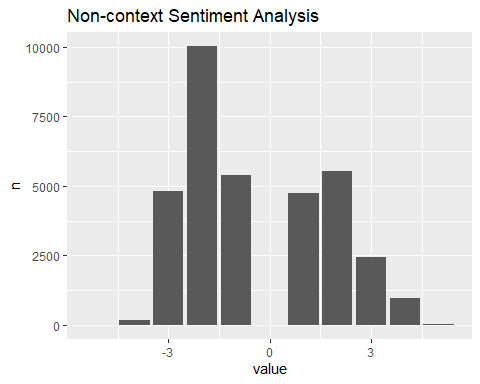
library(ggplot2)  
  
not\_words %>%  
 mutate(contribution = n \* value) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(n \* value, word2, fill = n \* value > 0)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = "Sentiment value \* number of occurrences. These sentiment should be reversed",  
 y = "Words preceded by \"not\"")



#Identify words that connect with not, no, never, without... then reverse the sentiment value of these words  
stop\_word2 = tibble(word = c(letters, LETTERS, "oh", 'just'))  
AFINN <- get\_sentiments("afinn")  
  
sentiment1 = df %>%  
 unnest\_tokens(input = highlights, output = word) %>%   
 anti\_join(get\_stopwords()) %>%   
 anti\_join(stop\_word2) %>%   
 inner\_join(get\_sentiments("afinn")) %>%   
 filter(!is.na(value)) %>%   
 count(value)

## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`  
## Joining with `by = join\_by(word)`

sentiment1 %>%   
 ggplot(aes(x = value, y = n))+  
 geom\_col()+   
 labs(title = 'Non-context Sentiment Analysis')



negation\_words <- c("not", "no", "never", "without")  
  
sentiment2 = bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word1, word2, value, sort = TRUE) %>%   
 select(value, n) %>%   
 group\_by(value) %>%   
 summarise(n = sum(n)) %>%   
 arrange(value)  
  
  
ggplot()+  
 geom\_col(data = sentiment1, aes(x =value, y = n))+  
 geom\_col(data = sentiment2, aes(x =value, y = n), fill = 'red')+  
 labs(title = 'Context Sentiment Analysis',  
 caption = "Red bars present the reversed sentiment.")

