607 Wk 11 - Sentiment Analysis

Jennifer Abinette

2022-11-03

Background

In this assignment, you should start by getting the primary example code from chapter 2 working in an R Markdown document. You should provide a citation to this base code.

Text Mining with R: A Tidy Approach, Julia Silge and David Robinson. O'Reilly, 2017.

We will also be working with the following lexicons: * `AFINN` from Finn Årup Nielsen (http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010), * `bing` from Bing Liu and collaborators (https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html), and * `nrc` from Saif Mohammad and Peter Turney (http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm) nrc dataset was published in Saif M. Mohammad and Peter Turney. (2013), ``Crowdsourcing a Word-Emotion Association Lexicon.'' Computational Intelligence, 29(3): 436-465.

You're then asked to extend the code in two ways: • Work with a different corpus of your choosing, and • Incorporate at least one additional sentiment lexicon (possibly from another R package that you've found through research). As usual, please submit links to both an .Rmd file posted in your GitHub repository and to your code on rpubs.com. You may work on a small team on this assignment.

Chapter 2 Sentiment Analysis of Jane Austen works

```
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE)))) %>%
  ungroup() %>%
  unnest_tokens(word, text)

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

tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 301 × 2
##    word          n
##    <chr>     <int>
##  1 good        359
##  2 friend      166
##  3 hope        143
##  4 happy       125
##  5 love        117
##  6 deal         92
##  7 found        92
##  8 present      89
##  9 kind         82
## 10 happiness    76
## # … with 291 more rows
```

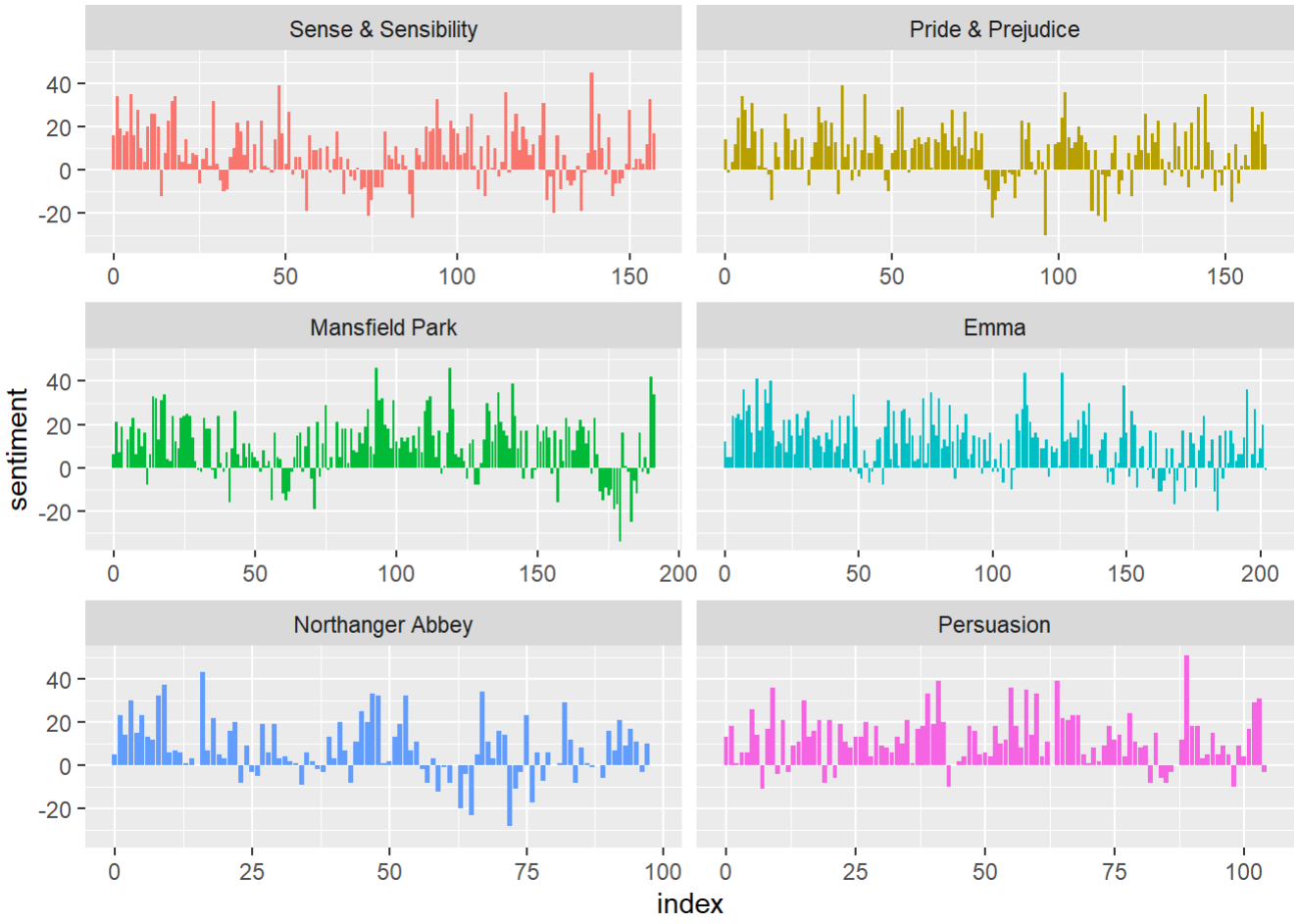
```
library(tidyr)

jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
library(ggplot2)

ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")
```




```
pride_prejudice <- tidy_books %>%
  filter(book == "Pride & Prejudice")

pride_prejudice
```

```
## # A tibble: 122,204 × 4
##    book               linenumber chapter word
##    <fct>                   <int>   <int> <chr>
##  1 Pride & Prejudice           1       0 pride
##  2 Pride & Prejudice           1       0 and
##  3 Pride & Prejudice           1       0 prejudice
##  4 Pride & Prejudice           3       0 by
##  5 Pride & Prejudice           3       0 jane
##  6 Pride & Prejudice           3       0 austen
##  7 Pride & Prejudice           7       1 chapter
##  8 Pride & Prejudice           7       1 1
##  9 Pride & Prejudice          10       1 it
## 10 Pride & Prejudice          10       1 is
## # … with 122,194 more rows
```

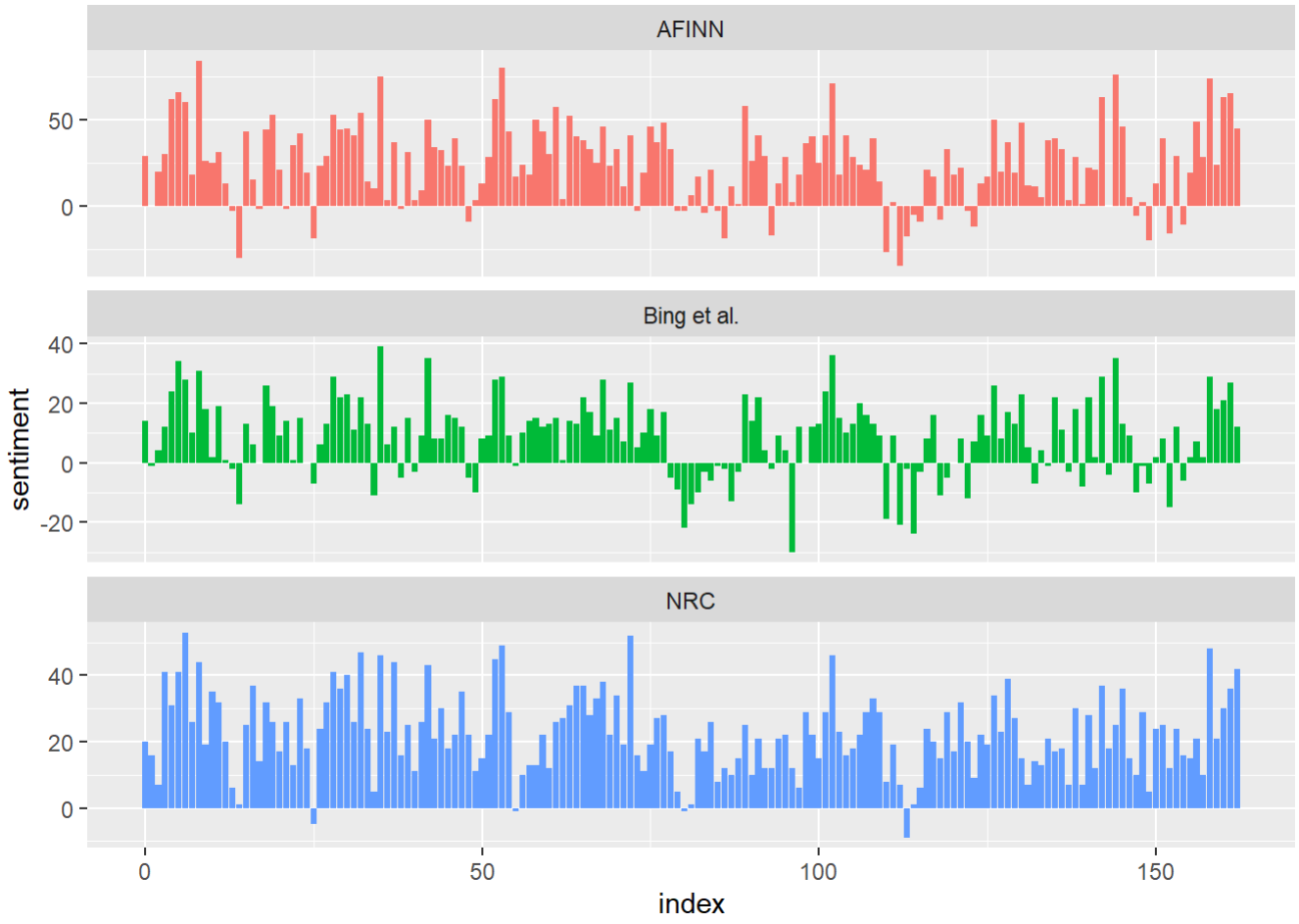
```
afinn <- pride_prejudice %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
bing_and_nrc <- bind_rows(
  pride_prejudice %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  pride_prejudice %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
bind_rows(afinn,
          bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



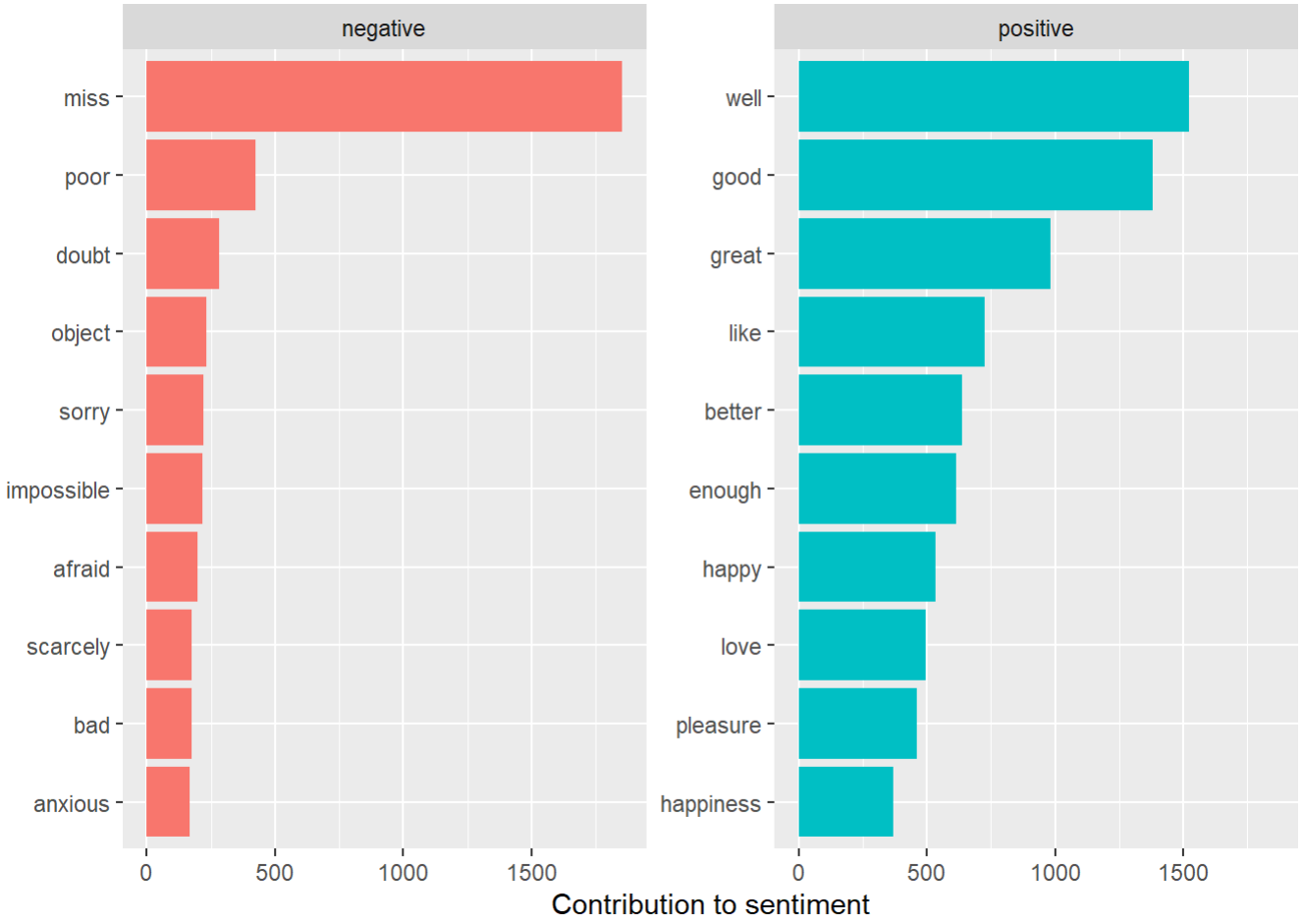

```
bing_word_counts <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

```
bing_word_counts
```

```
## # A tibble: 2,585 × 3
##    word      sentiment      n
##    <chr>     <chr>      <int>
##  1 miss      negative    1855
##  2 well      positive    1523
##  3 good      positive    1380
##  4 great     positive     981
##  5 like      positive     725
##  6 better    positive     639
##  7 enough    positive     613
##  8 happy     positive     534
##  9 love      positive     495
## 10 pleasure  positive     462
## # … with 2,575 more rows
```

```
bing_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
```




```
custom_stop_words <- bind_rows(tibble(word = c("miss"),
                                      lexicon = c("custom")),
                               stop_words)

custom_stop_words
```

```
## # A tibble: 1,150 × 2
##    word        lexicon
##    <chr>       <chr>
##  1 miss        custom
##  2 a           SMART
##  3 a's         SMART
##  4 able        SMART
##  5 about       SMART
##  6 above       SMART
##  7 according   SMART
##  8 accordingly SMART
##  9 across      SMART
## 10 actually    SMART
## # … with 1,140 more rows
```

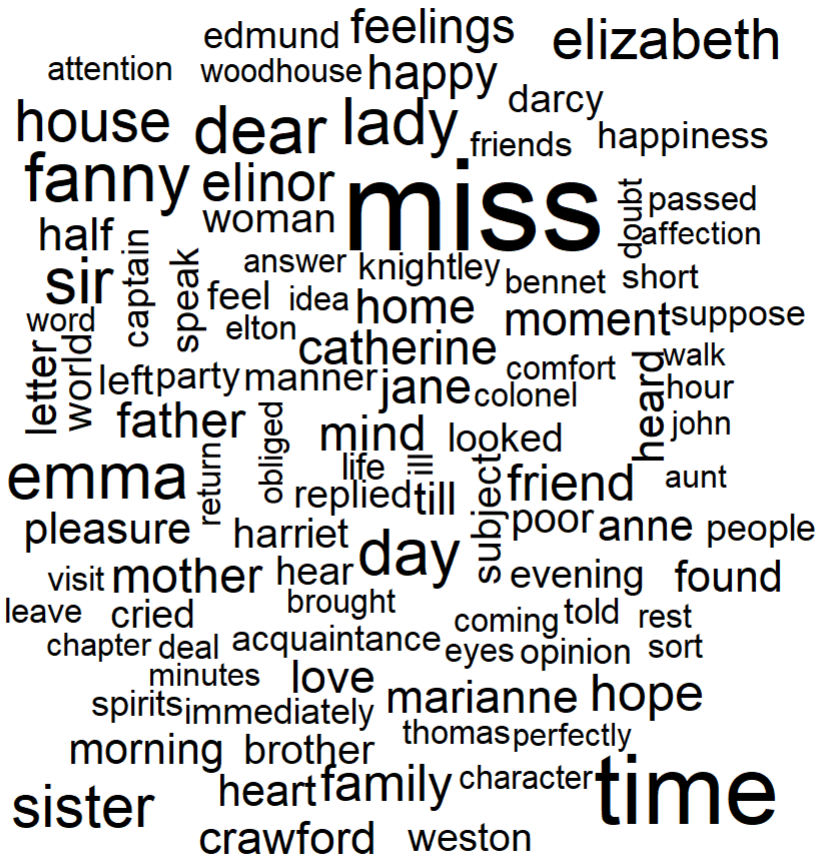
```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.2.2
```

```
## Loading required package: RColorBrewer
```

```
tidy_books %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

```
## Joining, by = "word"
```



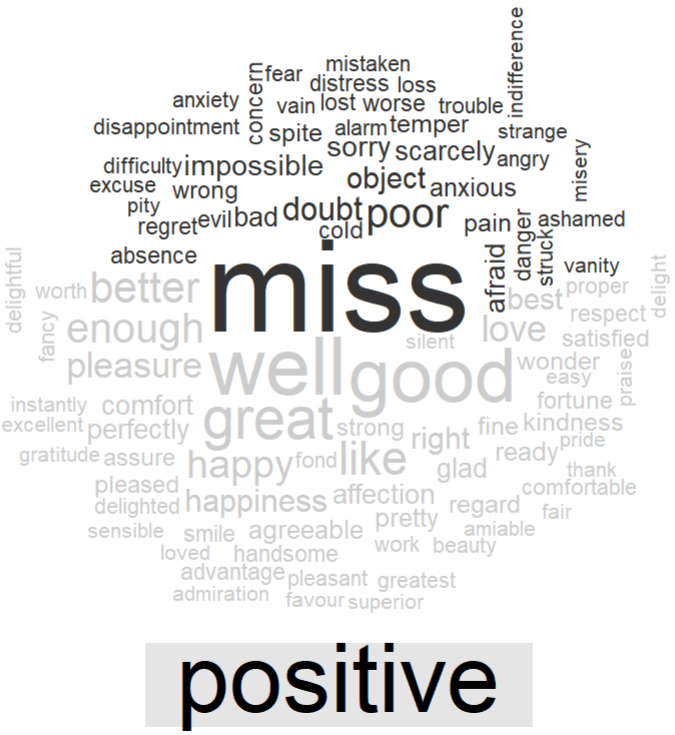

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                   max.words = 100)
```

```
## Joining, by = "word"
```



Sentiment Analysis of Right Troll Internet Research Agency (IRA) tweets

Data

We will be working with the 1st file of russian-troll-tweets from fivethirtyeight.com titled ‘IRAhandle_tweets_1.csv’. The full dataset includes 3 million Russian troll tweets from accounts associated with the Internet Research Agency based in St. Petersburg, Russia that “campaigned to sow disinformation and discord into American politics via social media.”

Original Data: https://github.com/fivethirtyeight/russian-troll-tweets (https://github.com/fivethirtyeight/russian-troll-tweets) Featured Article: https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/ (https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/)

Original data has been adjusted by: a) Convert the csv file to xlsx and save on my directory. b) File be too large so made the following adjustments: -Removed tweets not in RightTroll account category, language was not English, and publish_date was before 2016 election was called on November 9 -Removed all columns except author, publish_date and content c) Saved in directory to load into R # Find adjusted dataset on Github at https://github.com/JAbinette/CUNY-607-DataAcquisition/blob/main/Wk%2011%20-%20IRAhandle_tweets_1%20-Filtered%20to%20RightTroll%20ONLY.xlsx (https://github.com/JAbinette/CUNY-607-DataAcquisition/blob/main/Wk%2011%20-%20IRAhandle_tweets_1%20-Filtered%20to%20RightTroll%20ONLY.xlsx)

```
library(readxl)
# Set path to excel spreadsheet
path = "Wk 11 - IRAhandle_tweets_1 -Filtered to RightTroll ONLY.xlsx"
IRAtweet1 <- read_excel(path)
as_tibble(IRAtweet1)
```

```
## # A tibble: 34,749 × 3
##    author      content                                        publish_date
##    <chr>       <chr>                                          <dttm>
##  1 1ERIK_LEE   Why is someone even against the #petition? I… 2015-09-23 09:02:00
##  2 1ERIK_LEE   Itâ€™s reasonable to ban firearms sales in #… 2015-09-23 09:03:00
##  3 4EVER_SUSAN #Raiders defense playing hungry .. Bending a… 2015-12-13 22:52:00
##  4 4EVER_SUSAN Let's go offense !!!! Start Up the #Carr !!!… 2015-12-13 22:52:00
##  5 4EVER_SUSAN I was shocked and heartbroken when @CBS canc… 2015-12-14 20:34:00
##  6 4EVER_SUSAN I used to call Eden Hazard 'overrated' as a … 2015-12-14 20:34:00
##  7 4EVER_SUSAN The Holidays are in full swing.  Need gift i… 2015-12-14 20:34:00
##  8 4EVER_SUSAN My mum had no electricity at here's so she h… 2015-12-14 20:34:00
##  9 4EVER_SUSAN VOTE for @Grandfathered at @peopleschoice aw… 2015-12-14 20:34:00
## 10 4EVER_SUSAN Whoâ€™s ready for tWitch & Allison's Dance C… 2015-12-14 20:35:00
## # … with 34,739 more rows
```

Analysis Based on Unigrams - afinn, bing & nrc

```
# Tidy dataset so the tweet content is one word per line
IRAtweet2 <- IRAtweet1 %>%
    group_by(author) %>%
  mutate(
    linenumber = row_number()) %>%
  ungroup() %>%
  unnest_tokens(word, content)

as_tibble(IRAtweet2)
```

```
## # A tibble: 545,985 × 4
##    author    publish_date        linenumber word
##    <chr>     <dttm>                   <int> <chr>
##  1 1ERIK_LEE 2015-09-23 09:02:00          1 why
##  2 1ERIK_LEE 2015-09-23 09:02:00          1 is
##  3 1ERIK_LEE 2015-09-23 09:02:00          1 someone
##  4 1ERIK_LEE 2015-09-23 09:02:00          1 even
##  5 1ERIK_LEE 2015-09-23 09:02:00          1 against
##  6 1ERIK_LEE 2015-09-23 09:02:00          1 the
##  7 1ERIK_LEE 2015-09-23 09:02:00          1 petition
##  8 1ERIK_LEE 2015-09-23 09:02:00          1 i'll
##  9 1ERIK_LEE 2015-09-23 09:02:00          1 watch
## 10 1ERIK_LEE 2015-09-23 09:02:00          1 you
## # … with 545,975 more rows
```

Conduct the sentiment analyses by word and plot results together

```
# Conduct AFINN sentiment analysis
afinnIRA <- IRAtweet2 %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
# BING and NRC
bingIR_nrcIRA <- bind_rows(
  IRAtweet2 %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  IRAtweet2 %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
bind_rows(afinnIRA,
          bingIR_nrcIRA) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```

We can see from the graphs above that the sentiment analyses using AFINN and Bing lexicons found that in general the tweets were negative with only few instances of positive instances whereas nrc lexicon fluctuated between the two.

Let's take a look the words most frequently used by sentiment from the bing lexicon

```
bingIRA_counts <- IRAtweet2 %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining, by = "word"
```

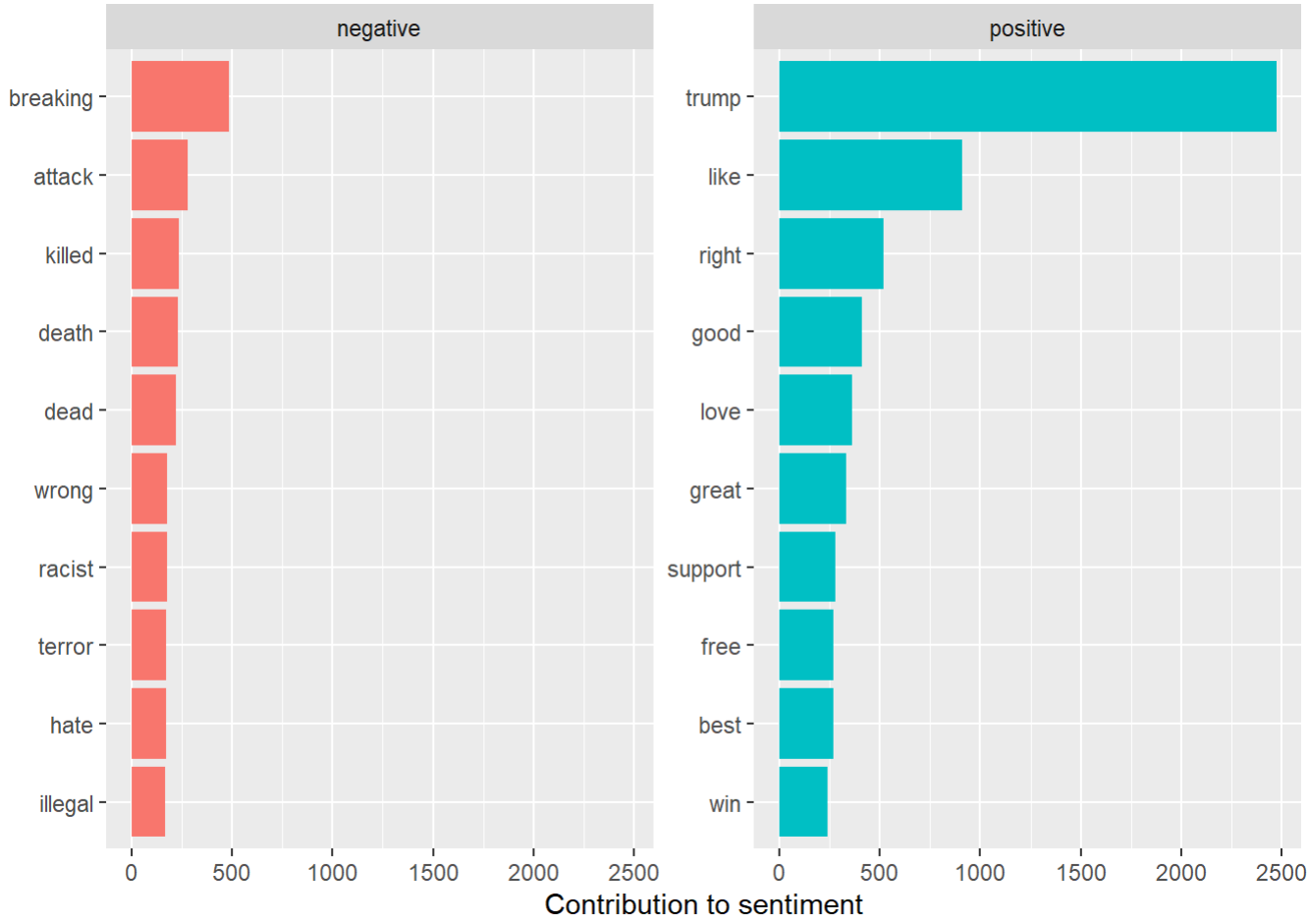
```
as_tibble(bingIRA_counts)
```

```
## # A tibble: 2,905 × 3
##    word     sentiment     n
##    <chr>    <chr>     <int>
##  1 trump    positive   2472
##  2 like     positive    908
##  3 right    positive    518
##  4 breaking negative    487
##  5 good     positive    411
##  6 love     positive    362
##  7 great    positive    335
##  8 attack   negative    282
##  9 support  positive    281
## 10 best     positive    270
## # … with 2,895 more rows
```

We can see that our sentiment analyses are highly affected by the word trump, which is used more than 2.5 times as much as the next highest word breaking. This has definitely affected our analyses as a Right Troll account publishing tweets from 2015 to November 9, 2016 is very likely to be referring to former President Donald Trump. If we did exclude trump from the lexicons, we would see the analyses become even more negative.

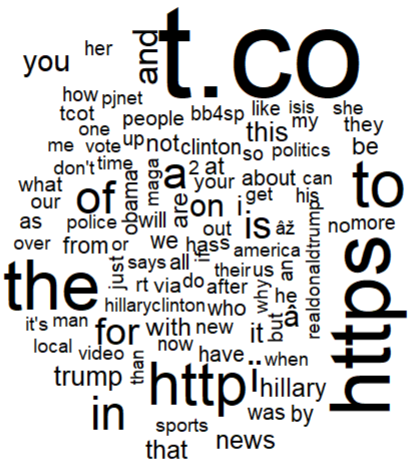
Show top 10 Positive and Negative words from Bing Sentiment Analysis

```
bingIRA_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
```



Build Wordcloud

```
IRAtweet2 %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



Sentiment Analysis by Tweet

Using the SentimentAnalysis package, we will use the psychological Harvard-IV dictionary, which is a general purpose dictionary to analyze whether a tweet is positive or negative. Unfortunately, R is not able to process all 113K tweets at once so we will need to split it.

SentimentAnalysis Vignette: https://cran.r-project.org/web/packages/SentimentAnalysis/vignettes/SentimentAnalysis.html#built-in-dictionaries (https://cran.r-project.org/web/packages/SentimentAnalysis/vignettes/SentimentAnalysis.html#built-in-dictionaries)

```
# Load dplyr
library(dplyr)

# Group by count using dplyr
agg_tbl <- IRAtweet1 %>%
  group_by(author) %>%
  summarise(total_count=n(),
            .groups = 'drop') %>%
  arrange(desc(total_count))
agg_tbl
```

```
## # A tibble: 42 × 2
##    author          total_count
##    <chr>                 <int>
##  1 ARM_2_ALAN            14530
##  2 AMELIEBALDWIN         13653
##  3 ARCHIEOLIVERS          1242
##  4 ALDRICH420             1090
##  5 ALBERTMORENMORE        1084
##  6 AMERICANALBERT          414
##  7 ALFREDTHREE             222
##  8 AUSTINLOVESBEER         204
##  9 AMIYAHSAMUELS           178
## 10 ADDIE_HOL               145
## # … with 32 more rows
```

Split data into 3 subsets for the Sentiment Analysis

```
sub1 <- subset(IRAtweet1, author == "ARM_2_ALAN")
sub2 <- subset(IRAtweet1, author == "AMELIEBALDWIN")
sub3 <- subset(IRAtweet1, author != "ARM_2_ALAN" & author != "AMELIEBALDWIN")
```

Conduct Sentiment Analysis of each tweet

```
library(SentimentAnalysis)
```

```
## Warning: package 'SentimentAnalysis' was built under R version 4.2.2
```

```
##
## Attaching package: 'SentimentAnalysis'
```

```
## The following object is masked from 'package:base':
##
##     write
```

```
data(DictionaryGI)
# Analyze sentiment
sentiment1 <- analyzeSentiment(sub1$content,
                               rules=list("SentimentGI"=list(ruleSentiment, loadDictionaryGI())))
sentiment2 <- analyzeSentiment(sub2$content,
                               rules=list("SentimentGI"=list(ruleSentiment, loadDictionaryGI())))
sentiment3 <- analyzeSentiment(sub3$content,
                               rules=list("SentimentGI"=list(ruleSentiment, loadDictionaryGI())))
```

Combine results and then convert to Binary Response to see overall were the tweets as a whole positive or negative.

```
# Combine
sentiment_all <- rbind(sentiment1, sentiment2, sentiment3)

# Count positive and negative tweets
table(convertToBinaryResponse(sentiment_all$SentimentGI))
```

```
##
## negative positive
##    11642    23105
```

Conclusions:

As we can see there were mixed results based on the lexicon used to conduct the sentiment analysis. The analyses of each word using AFINN or Bing yielded mostly negative results, whereas nrc was a mix. Lastly, we can see directly above that analyzing the sentiment per tweet using the Harvard-IV dictionary showed there were just under twice as many positive tweets than negative. We can clearly see more analyses is needed to satisfactorily conclude one way or another and should exclude the word trump from the lexicons.