Chapter 5

John Peach

3/30/2021

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
data("AssociatedPress", package = 'topicmodels')
AssociatedPress
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
terms <- Terms(AssociatedPress)
head(terms)
```

```
## [1] "aaron"      "abandon"    "abandoned"  "abandoning" "abbott"
## [6] "abboud"
```

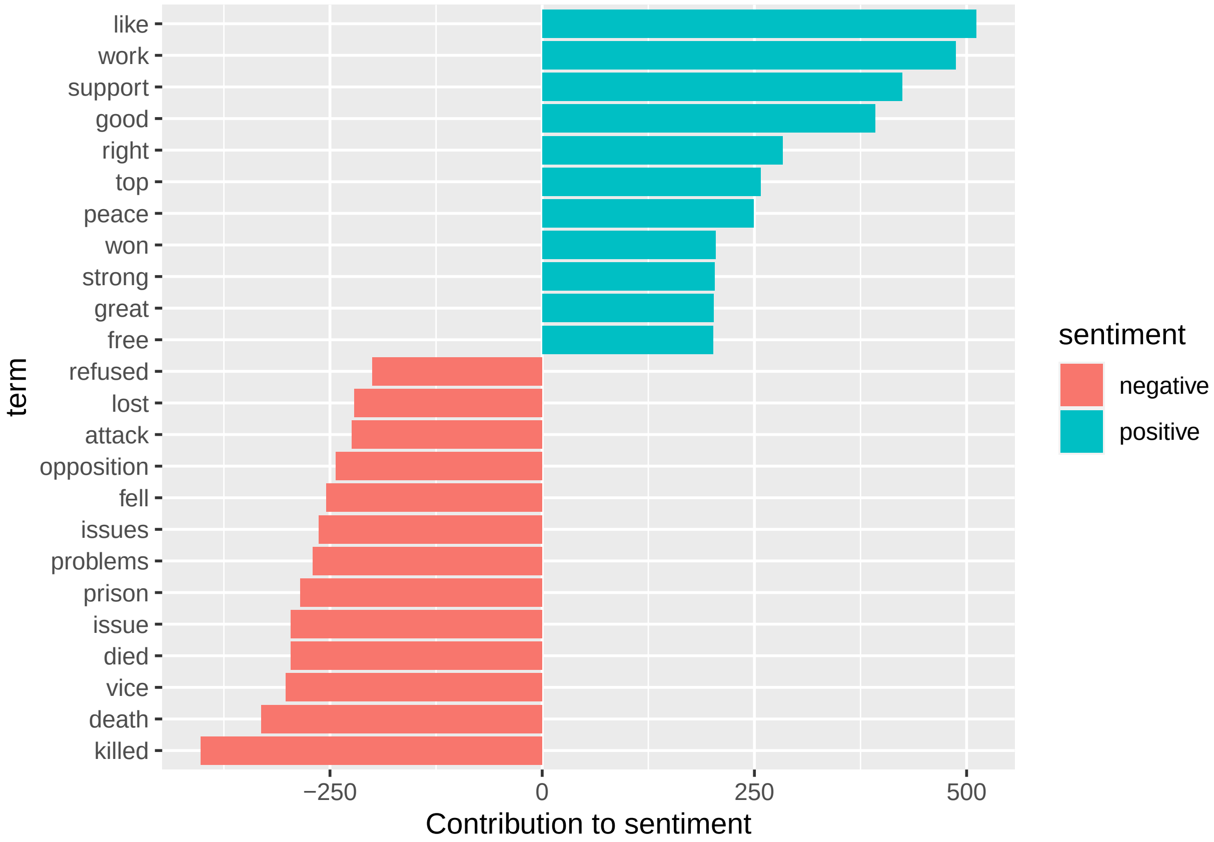
```
library(dplyr)
library(tidytext)

ap_td <- tidy(AssociatedPress)
ap_td
```

```
## # A tibble: 302,031 x 3
##    document term       count
##       <int> <chr>      <dbl>
##  1        1 adding         1
##  2        1 adult          2
##  3        1 ago            1
##  4        1 alcohol        1
##  5        1 allegedly      1
##  6        1 allen          1
##  7        1 apparently     2
##  8        1 appeared       1
##  9        1 arrested       1
## 10        1 assault        1
## # ... with 302,021 more rows
```

```
ap_sentiments <- ap_td %>%
  inner_join(get_sentiments('bing'), by = c(term = 'word'))
ap_sentiments
```

```
## # A tibble: 30,094 x 4
##    document term     count sentiment
##       <int> <chr>    <dbl> <chr>
##  1        1 assault      1 negative
##  2        1 complex      1 negative
##  3        1 death        1 negative
##  4        1 died         1 negative
##  5        1 good         2 positive
##  6        1 illness      1 negative
##  7        1 killed       2 negative
##  8        1 like         2 positive
##  9        1 liked        1 positive
## 10        1 miracle      1 positive
## # ... with 30,084 more rows
```

```
library(ggplot2)

ap_sentiments %>%
  count(sentiment, term, wt = count) %>%
  ungroup() %>%
  dplyr::filter(n >= 200) %>%
  mutate(n = ifelse(sentiment == 'negative', -n, n)) %>%
  mutate(term = reorder(term, n)) %>%
  ggplot(aes(term, n, fill = sentiment)) +
    geom_bar(stat = 'identity') +
    ylab('Contribution to sentiment') +
    coord_flip()
```



```
library(methods)

data("data_corpus_inaugural", package = 'quanteda')
inaug_dfm <- quanteda::dfm(data_corpus_inaugural, verbose = FALSE)
inaug_dfm
```

```
## Document-feature matrix of: 58 documents, 9,360 features (91.8% sparse) and 4 docvars.
##                     features
## docs                 fellow-citizens  of the senate and house representatives :
##   1789-Washington                  1  71 116      1  48     2               2 1
##   1793-Washington                  0  11  13      0   2     0               0 1
##   1797-Adams                       3 140 163      1 130     0               2 0
##   1801-Jefferson                   2 104 130      0  81     0               0 1
##   1805-Jefferson                   0 101 143      0  93     0               0 0
##   1809-Madison                     1  69 104      0  43     0               0 0
##                     features
## docs                 among vicissitudes
##   1789-Washington        1            1
##   1793-Washington        0            0
##   1797-Adams             4            0
##   1801-Jefferson         1            0
##   1805-Jefferson         7            0
##   1809-Madison           0            0
## [ reached max_ndoc ... 52 more documents, reached max_nfeat ... 9,350 more features ]
```

```
inaug_td <- tidy(inaug_dfm)
inaug_td
```

```
## # A tibble: 44,710 x 3
##    document        term            count
##    <chr>           <chr>           <dbl>
##  1 1789-Washington fellow-citizens     1
##  2 1797-Adams      fellow-citizens     3
##  3 1801-Jefferson  fellow-citizens     2
##  4 1809-Madison    fellow-citizens     1
##  5 1813-Madison    fellow-citizens     1
##  6 1817-Monroe     fellow-citizens     5
##  7 1821-Monroe     fellow-citizens     1
##  8 1841-Harrison   fellow-citizens    11
##  9 1845-Polk       fellow-citizens     1
## 10 1849-Taylor     fellow-citizens     1
## # ... with 44,700 more rows
```

```
inaug_tf_idf <- inaug_td %>%
  bind_tf_idf(term, document, count) %>%
  arrange(desc(tf_idf))

inaug_tf_idf
```

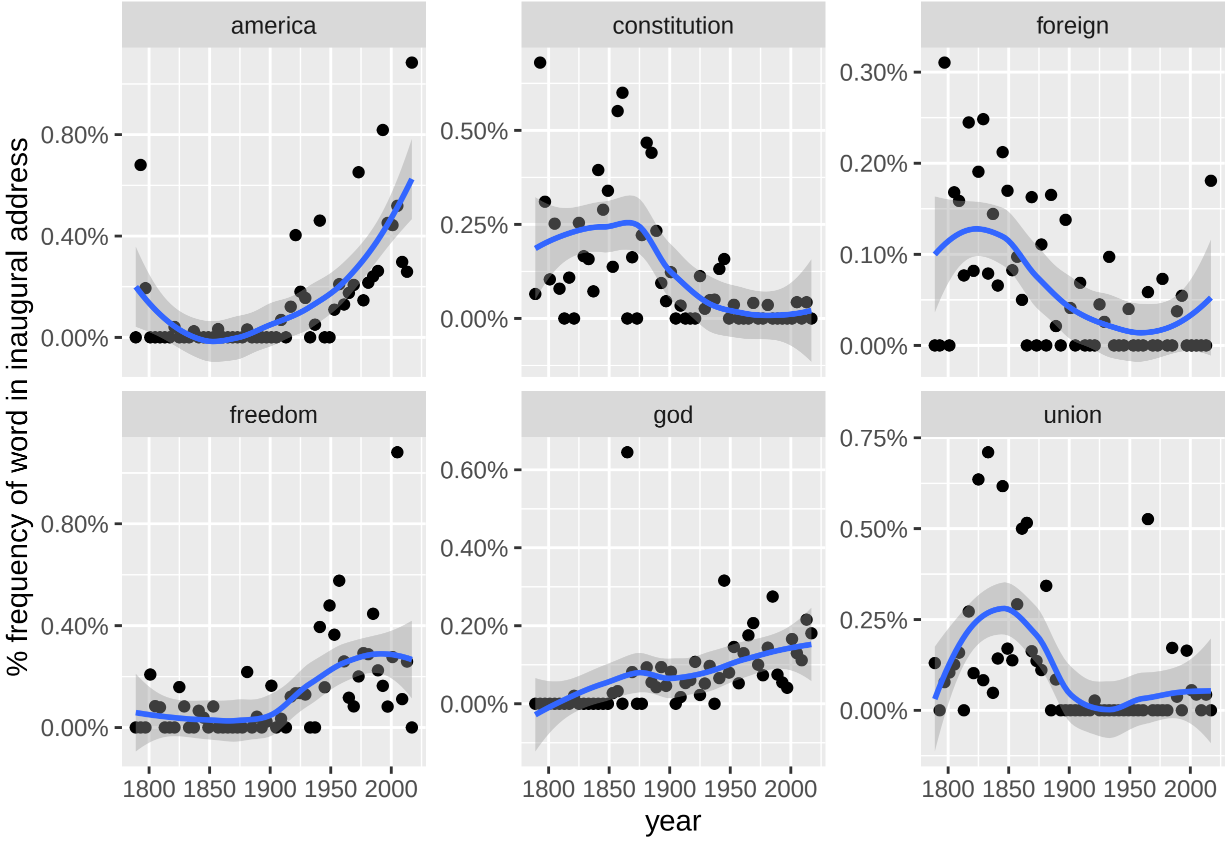
```
## # A tibble: 44,710 x 6
##    document        term        count      tf   idf tf_idf
##    <chr>           <chr>       <dbl>   <dbl> <dbl>  <dbl>
##  1 1793-Washington arrive          1 0.00680  4.06 0.0276
##  2 1793-Washington upbraidings     1 0.00680  4.06 0.0276
##  3 1793-Washington violated        1 0.00680  3.37 0.0229
##  4 1793-Washington willingly       1 0.00680  3.37 0.0229
##  5 1793-Washington incurring       1 0.00680  3.37 0.0229
##  6 1793-Washington previous        1 0.00680  2.96 0.0201
##  7 1793-Washington knowingly       1 0.00680  2.96 0.0201
##  8 1793-Washington injunctions     1 0.00680  2.96 0.0201
##  9 1793-Washington witnesses       1 0.00680  2.96 0.0201
## 10 1793-Washington besides         1 0.00680  2.67 0.0182
## # ... with 44,700 more rows
```

```
library(tidyr)

year_term_counts <- inaug_td %>%
  extract(document, "year", "(\\d+)", convert = TRUE) %>%
  complete(year, term, fill = list(count = 0)) %>%
  group_by(year) %>%
  mutate(year_total = sum(count))

year_term_counts
```

```
## # A tibble: 542,880 x 4
## # Groups:   year [58]
##     year term  count year_total
##    <int> <chr> <dbl>      <dbl>
##  1  1789 "-"       1       1537
##  2  1789 ","      70       1537
##  3  1789 ";"       8       1537
##  4  1789 ":"       1       1537
##  5  1789 "!"       0       1537
##  6  1789 "?"       0       1537
##  7  1789 "."      23       1537
##  8  1789 "'"       0       1537
##  9  1789 "\""      2       1537
## 10  1789 "("       1       1537
## # ... with 542,870 more rows
```

```
library(ggplot2)
year_term_counts %>%
  dplyr::filter(term %in% c("god", "america", "foreign", "union",
                            "constitution", "freedom")) %>%
  ggplot(aes(year, count / year_total)) +
    geom_point() +
    geom_smooth() +
    facet_wrap(~term, scales = "free_y") +
    scale_y_continuous(labels = scales::percent_format()) +
    ylab("% frequency of word in inaugural address")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
ap_td %>%
  cast_dtm(document, term, count)
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
m <- ap_td %>%
  cast_sparse(document, term, count)

class(m)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

```
dim(m)
```

```
## [1]  2246 10473
```

```
library(janeaustenr)

austen_dtm <- austen_books() %>%
  unnest_tokens(word, text) %>%
  count(book, word) %>%
  cast_dtm(book, word, n)

austen_dtm
```

```
## <<DocumentTermMatrix (documents: 6, terms: 14520)>>
## Non-/sparse entries: 40379/46741
## Sparsity           : 54%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

```
library(tm)
data("acq")
acq
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 50
```

```
acq_td <- tidy(acq)
acq_td
```

```
## # A tibble: 50 x 16
##    author datetimestamp       description heading id    language origin topics
##    <chr>  <dttm>              <chr>       <chr>   <chr> <chr>    <chr>  <chr>
##   1  <NA>  1987-02-26 07:18:06 ""          COMPUT~ 10    en       Reute~ YES
##   2  <NA>  1987-02-26 07:19:15 ""          OHIO M~ 12    en       Reute~ YES
##   3  <NA>  1987-02-26 07:49:56 ""          MCLEAN~ 44    en       Reute~ YES
##   4 By Ca~ 1987-02-26 07:51:17 ""          CHEMLA~ 45    en       Reute~ YES
##   5  <NA>  1987-02-26 08:08:33 ""          <COFAB~ 68    en       Reute~ YES
##   6  <NA>  1987-02-26 08:32:37 ""          INVEST~ 96    en       Reute~ YES
##   7 By Pa~ 1987-02-26 08:43:13 ""          AMERIC~ 110   en       Reute~ YES
##  8 <NA>   1987-02-26 08:59:25 ""          HONG K~ 125   en       Reute~ YES
##  9 <NA>   1987-02-26 09:01:28 ""          LIEBER~ 128   en       Reute~ YES
## 10 <NA>   1987-02-26 09:08:27 ""          GULF A~ 134   en       Reute~ YES
## # ... with 40 more rows, and 8 more variables: lewissplit <chr>,
## #   cgisplit <chr>, oldid <chr>, places <named list>, people <lgl>, orgs <lgl>,
## #   exchanges <lgl>, text <chr>
```

```
acq_tokens <- acq_td %>%
  select(-places) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = "word")
```

```
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
```

```
acq_tokens %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 1,566 x 2
##    word         n
##    <chr>    <int>
##  1 dlrs       100
##  2 pct         70
##  3 mln         65
##  4 company     63
##  5 shares      52
##  6 reuter      50
##  7 stock       46
##  8 offer       34
##  9 share       34
## 10 american    28
## # ... with 1,556 more rows
```

```
acq_tokens %>%
  count(id, word) %>%
  bind_tf_idf(word, id, n) %>%
  arrange(desc(tf_idf))
```

```
## # A tibble: 2,853 x 6
##    id    word          n     tf   idf tf_idf
##    <chr> <chr>    <int>  <dbl> <dbl>  <dbl>
##  1 186   groupe       2 0.133   3.91  0.522
##  2 128   liebert      3 0.130   3.91  0.510
##  3 474   esselte      5 0.109   3.91  0.425
##  4 371   burdett      6 0.103   3.91  0.405
##  5 442   hazleton     4 0.103   3.91  0.401
##  6 199   circuit      5 0.102   3.91  0.399
##  7 162   suffield     2 0.1     3.91  0.391
##  8 498   west         3 0.1     3.91  0.391
##  9 441   rmj          8 0.121   3.22  0.390
## 10 467   nursery      3 0.0968  3.91  0.379
## # ... with 2,843 more rows
```

```
library(tm.plugin.webmining)
```

```
##
## Attaching package: 'tm.plugin.webmining'
## The following object is masked from 'package:tidyr':
##
##     extract

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

```r
library(purrr)
library(tidyverse)

company <- c("Microsoft", "Apple", "Google", "Amazon", "Facebook",
             "Twitter", "IBM", "Yahoo", "Netflix")
symbol <- c("MSFT", "AAPL", "GOOG", "AMZN", "FB", "TWTR", "IBM", "YHOO", "NFLX")

# Use YahooNewsSource instead of GoogleFinanceSource
download_article <- function(symbol) {
  WebCorpus(YahooNewsSource(paste0("NASDAQ:", symbol)))
}

stock_articles <- data_frame(company = company,
                             symbol = symbol) %>%
  mutate(corpus = map(symbol, download_article))
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```r
# This code does not work. I think there is a bug in the tidy verse
stock_tokens <- stock_articles %>%
  unnest(map(corpus, tidy)) %>%
  unnest_tokens(word, text) %>%
  select(company, datetimestamp, word, id, heading)

stock_tokens
```

```r
library(stringr)

stock_tf_idf <- stock_tokens %>%
  count(company, word) %>%
  dplyr::filter(!str_detect(word, "\\d+")) %>%
  bind_tf_idf(word, company, n) %>%
  arrange(-tf_idf)
```

```r
stock_tokens %>%
  anti_join(stop_words, by = "word") %>%
  count(word, id, sort = TRUE) %>%
  inner_join(get_sentiments('afinn'), by = "word") %>%
  group_by(word) %>%
  summarise(contribution = sum(n * score)) %>%
  top_n(12, abs(contribution)) %>%
  mutate(word = reorder(word, contribution)) %>%
  ggplot(aes(word, contribution)) +
    geom_col() +
    coord_flip() +
    labs(y = 'Frequency of word * AFINN score')
```

```
stock_tokens %>%
  count(word) %>%
  inner_join(get_sentiments('loughran'), by = 'word') %>%
  group_by(sentiment) %>%
  top_n(5, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
    geom_col() +
    coord_flip() +
    facet_wrap(~ sentiment, scales = 'free') +
    ylab("Frequency of this word in the recent financial articles")
```

```
stock_sentiment_count <- stock_tokens %>%
  inner_join(get_sentiments('loughran'), by = 'word') %>%
  count(sentiment, company) %>%
  spread(sentiment, n, fill = 0)

stock_sentiment_count
```

```
stock_sentiment_count %>%
  mutate(score = (positive - negative) / (positive + negative)) %>%
  mutate(company = reorder(company, score)) %>%
  ggplot(ase(company, score, fill = score > 0)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    labs(x = "Company", y = "Positivity score among 20 recent news articles")
```