Text Analysis

Maggie Nead

7/12/2020

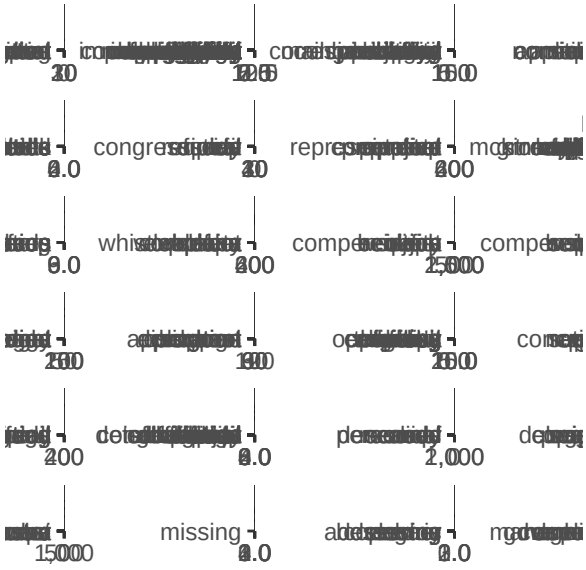
##Packages

##data clean

```
data %<>%
  filter(is.na(TYPE), !is.na(SUBJECT))

# Make a single string of stopwords separated by regex "OR" ("|")
stopwords <- str_c(stop_words$word, collapse = " | ")
# Add to the list of things to exclude
stopwords <- paste("[0-9]|", stopwords, "| senator", "| representative ", "| write", "| writes", "| wri

data$SUBJECT %<>%
  # To lower case
  tolower() %>%
  # Remove stopwords
  str_replace(stopwords, " ") %>%
  # Remove numbers
  str_replace_all("[0-9]", "")
```

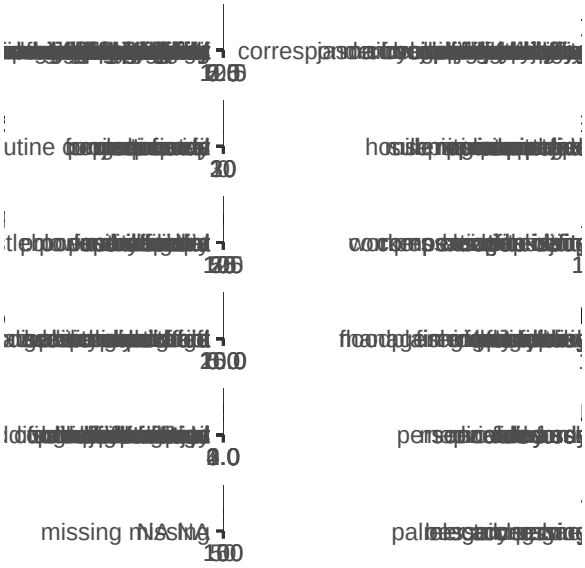
##word counts

```
words <- data %>%
  unnest_tokens(word, SUBJECT) %>%
  filter(!(word %in% stop_words$word)) %>%
  group_by(agency) %>%
  count(word, sort = TRUE) %>%
  top_n(10) %>%
  mutate(word = fct_inorder(word))

ggplot(words, aes(x = fct_rev(word), y = n)) +
  geom_col() +
  coord_flip() +
  scale_y_continuous(labels = scales::comma) +
  labs(y = "Count", x = NULL, title = "10 most frequent words") +
  facet_wrap("agency", scales = "free")
```



##bigrams

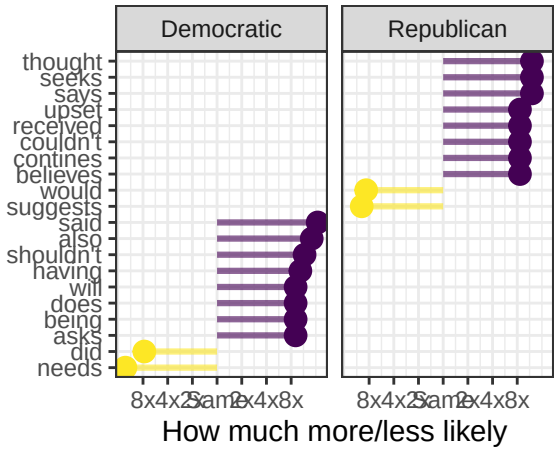
```
bigrams <- data %>%
  group_by(agency) %>%
  unnest_tokens(bigram, SUBJECT, token = "ngrams", n = 2) %>%
  # Split the bigram column into two columns
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word) %>%
  # Put the two word columns back together
  unite(bigram, word1, word2, sep = " ") %>%
  count(bigram, sort = TRUE) %>%
  top_n(10)

ggplot(bigrams, aes(x = reorder(bigram, n), y = n)) +
  geom_col() +
  coord_flip() +
  scale_y_continuous(labels = scales::comma) +
  labs(y = "Count", x = NULL, title = "10 most frequent word pairs") +
  facet_wrap("agency", scales = "free")
```



##bigrams and probability

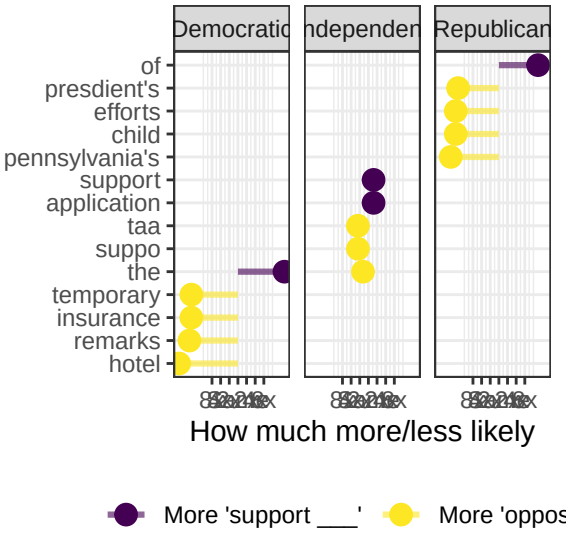
```
pronouns <- c("he", "she")

bigram_binary_counts <- data %>%
  group_by(party_name) %>%
  unnest_tokens(bigram, SUBJECT, token = "ngrams", n = 2) %>%
  # count(bigram, sort = TRUE) %>%
  # Split the bigram column into two columns
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  # Only choose rows where the first word is he or she
  filter(word1 %in% pronouns) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(total = n)

word_ratios <- bigram_binary_counts %>%
  # Spread out the word1 column so that there's a column named "he" and one named "she"
  spread(word1, total, fill = 0) %>%
  # Add 1 to each number so that logs work (just in case any are zero)
  mutate_if(is.numeric, funs((. + 1) / sum(. + 1))) %>%
  # Create a new column that is the logged ratio of the she counts to he counts
  mutate(logratio = log2(she / he)) %>%
  # take the absolute value
  mutate(abslogratio = abs(logratio))

word_ratios %>%
  top_n(10, abslogratio) %>%
  mutate(word = reorder(word2, logratio)) %>%
  ggplot(aes(word, logratio, color = logratio < 0)) +
  geom_segment(aes(x = word, xend = word,
                   y = 0, yend = logratio),
               size = 1.1, alpha = 0.6) +
  geom_point(size = 3.5) +
  coord_flip() +
  labs(y = "How much more/less likely", x = NULL) +
  scale_color_discrete(name = "", labels = c("More 'she'", "More 'he'")) +
```

```
  scale_y_continuous(breaks = seq(-3, 3), labels = c("8x", "4x", "2x", "Same", "2x", "4x", "8x")) +
  theme(legend.position = "bottom") +
  facet_grid(. ~ party_name)
```




```
by <- c("support", "oppose")

bigram_support_oppose_counts <- data %>%
  # Regular expression (regex) to match any suffix
  mutate(SUBJECT = str_replace_all(SUBJECT, "support[a-z]*", "support") ) %>%
  mutate(SUBJECT = str_replace_all(SUBJECT, "oppos[a-z]*", "oppose") ) %>%
  group_by(party_name) %>%
  unnest_tokens(bigram, SUBJECT, token = "ngrams", n = 2) %>%
  # Split the bigram column into two columns
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  # Only choose rows where the first word is support or oppose
  filter(str_detect(word1, by)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(total = n)

word_ratios <- bigram_support_oppose_counts %>%
  group_by(party_name) %>%
  # Spread out the word1 column so that there's a column named "support" and one named "oppose"
  spread(word1, total, fill = 0) %>%
  # Add 1 to each number so that logs work (just in case any are zero)
  mutate_if(is.numeric, funs((. + 1) / sum(. + 1))) %>%
  # Create a new column that is the logged ratio of the she counts to he counts
  mutate(logratio = log2(support / oppose)) %>%
  # take the absolute value
  mutate(abslogratio = abs(logratio))

word_ratios %>%
  top_n(5, abslogratio) %>%
  ungroup() %>%
  group_by(party_name) %>%
  mutate(word = reorder(word2, logratio))  %>%
  ggplot( aes(word, logratio, color = logratio < 0)) +
```

```
  geom_segment(aes(x = word, xend = word,
                   y = 0, yend = logratio),
               size = 1.1, alpha = 0.6) +
  geom_point(size = 3.5) +
  coord_flip() +
  labs(y = "How much more/less likely", x = NULL) +
  scale_color_discrete(name = "", labels = c("More 'support ___'", "More 'oppose ___'")) +
  scale_y_continuous(breaks = seq(-3, 3),
                     labels = c("8x", "4x", "2x",
                                "Same", "2x", "4x", "8x")) +
  theme(legend.position = "bottom") +
  facet_grid(. ~ party_name)
```



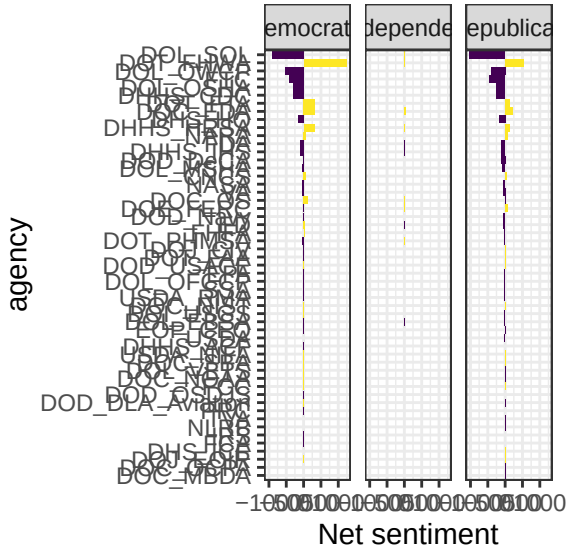
##sentiment analysis

```
# Dictionaries
get_sentiments("afinn") # Scoring system
```

```
## # A tibble: 2,477 x 2
##    word       value
##    <chr>      <dbl>
##  1 abandon       -2
##  2 abandoned     -2
##  3 abandons      -2
##  4 abducted      -2
##  5 abduction     -2
##  6 abductions    -2
##  7 abhor         -3
##  8 abhorred      -3
##  9 abhorrent     -3
## 10 abhors        -3
## # ... with 2,467 more rows
```

```
sentiment <- data %>%
  # Split into individual words
  unnest_tokens(word, SUBJECT) %>%
  # Join bing sentiment dicionary
  inner_join(get_sentiments("bing")) %>%
```

```
  # Count how many postive/negative words are in each chapter
  count(party_name, agency, sentiment) %>%
  # Spread the count into two columns named positive and negative
  spread(sentiment, n, fill = 0) %>%
  # Subtract the positive words from the negative words
  mutate(net_sentiment = positive - negative)

sentiment %>%
  ggplot() +
  aes(x = reorder(agency, abs(net_sentiment)),
      y = net_sentiment, fill = net_sentiment > 0) +
  geom_col() +
  guides(fill = FALSE) +
  coord_flip() +
  labs(x = "agency", y = "Net sentiment") +
  facet_wrap("party_name")
```




```
words <- data %>%
  unnest_tokens(word, SUBJECT) %>%
  group_by(agency) %>%
  count(word, sort = TRUE) %>%
  ungroup()
```