Sentiment_Analysis

Ruthvik Ravindra

3/28/2020

Include required libraries

```
library(readr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------- tidyvers
e 1.2.1 --
```

```
## v tibble  2.1.3     v purrr   0.3.3
## v tidyr   1.0.0     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------- tidyverse_conf
licts() --
## x dplyr::filter() masks stats::filter()
## x dbplyr::ident() masks dplyr::ident()
## x dplyr::lag()    masks stats::lag()
## x dbplyr::sql()   masks dplyr::sql()
```

```
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(tidytext)
library(tokenizers)
library(gutenbergr)
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.6.2
```

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

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

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

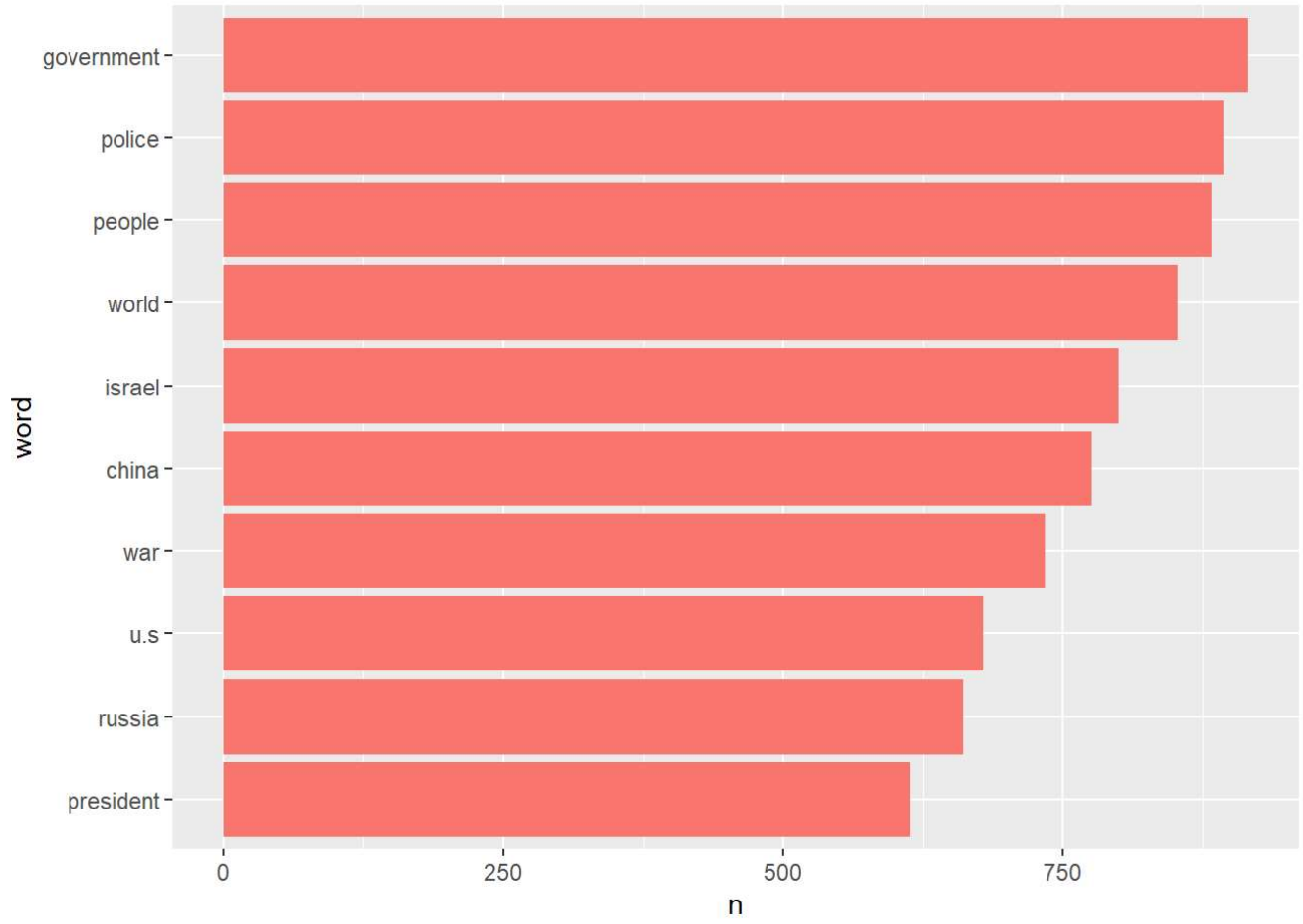
Down Market Analysis

```
speech <- read_lines("down_market.txt")

tspeech <- tibble(line=1:length(speech),text = speech)
```

Find Most Used Words

```
tspeech %>%
  unnest_tokens(word,text)%>%
anti_join(stop_words, by="word") %>%
count(word, sort=TRUE) %>%
filter(n > 300) %>%
mutate(word = reorder(word, n)) %>%
top_n(10)%>%
ggplot(aes(x=word, y=n,fill="red")) +
geom_col(show.legend = FALSE) +
coord_flip()
```

```
## Selecting by n
```



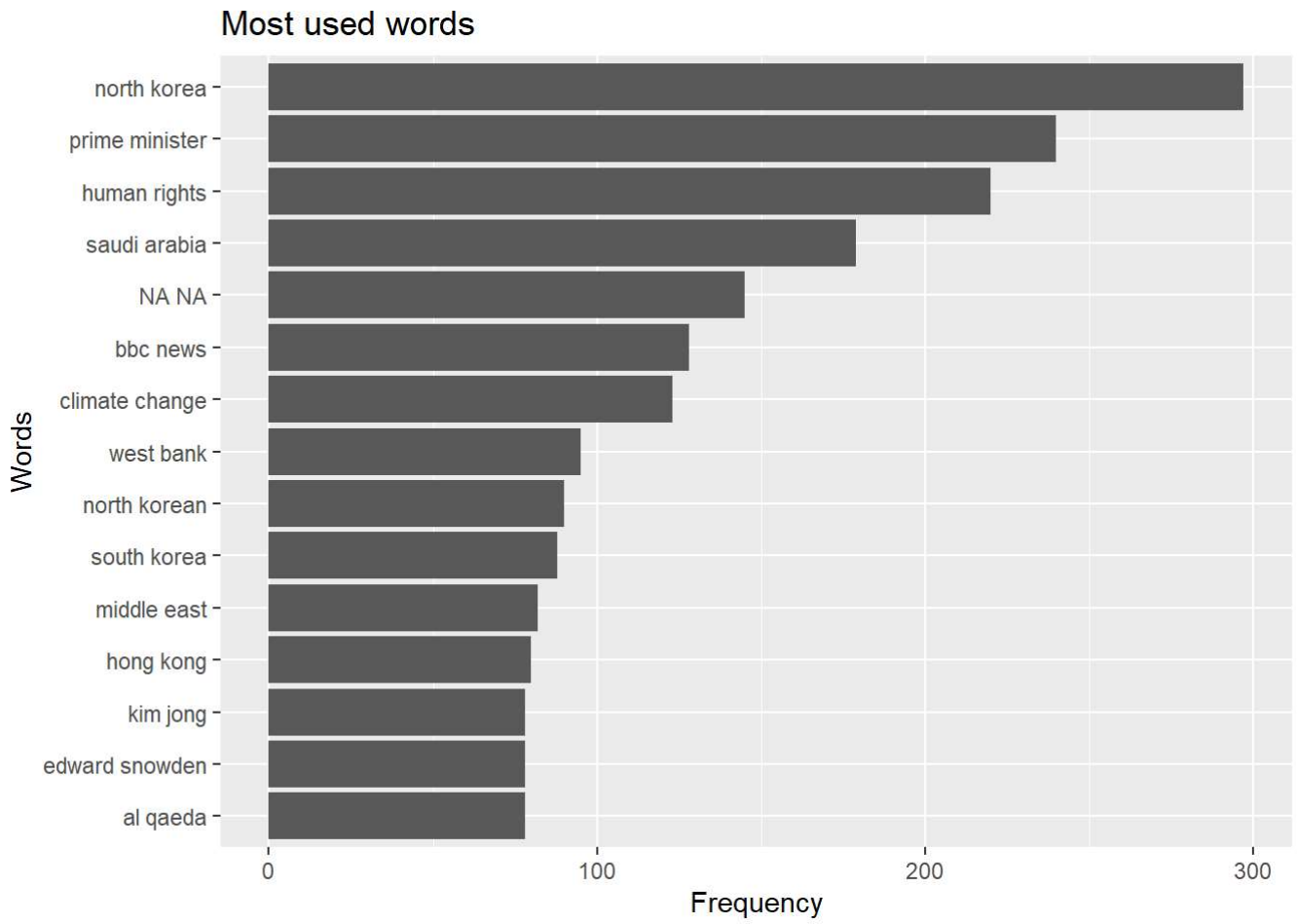
Find Most Used Bigrams

```
speech_bigrams <- unnest_tokens(tspeech, bigram, text,token = "ngrams", n=2)
speech_bigrams
```

```
## # A tibble: 412,072 x 2
##     line bigram
##    <int> <chr>
##  1     1 b georgia
##  2     1 georgia downs
##  3     1 downs two
##  4     1 two russian
##  5     1 russian warplanes
##  6     1 warplanes as
##  7     1 as countries
##  8     1 countries move
##  9     1 move to
## 10     1 to brink
## # ... with 412,062 more rows
```

```
speech_bigrams <- speech_bigrams %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_stop <- tibble(word = c("applause"))



speech_bigrams <- speech_bigrams %>%
  filter(!word1 %in% stop_words$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word1 %in% speech_stop$word)%>%
  filter(!word2 %in% speech_stop$word)

speech_negation <- tibble(word = c("never",
                                   "no",
                                   "without",
                                   "not"))

speech_bigrams <- speech_bigrams %>%
  filter(!word1 %in% speech_negation$word)

speech_bigrams %>%
  count(word1,word2,sort=TRUE) %>%
  unite(bigram,c(word1,word2),sep=" ")%>%
  top_n(15) %>%
  mutate(word= reorder(bigram,n))%>%
  ggplot(aes(x=word,y=n)) +
  geom_col() +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Most used words") +
  coord_flip()
```

```
## Selecting by n
```



Bigram Connector graph

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.6.3
```

```
##
## Attaching package: 'igraph'
```

```
## 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:dplyr':
##
##     as_data_frame, groups, union
```

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

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

```
library(ggraph)
```

```
## Warning: package 'ggraph' was built under R version 3.6.3
```

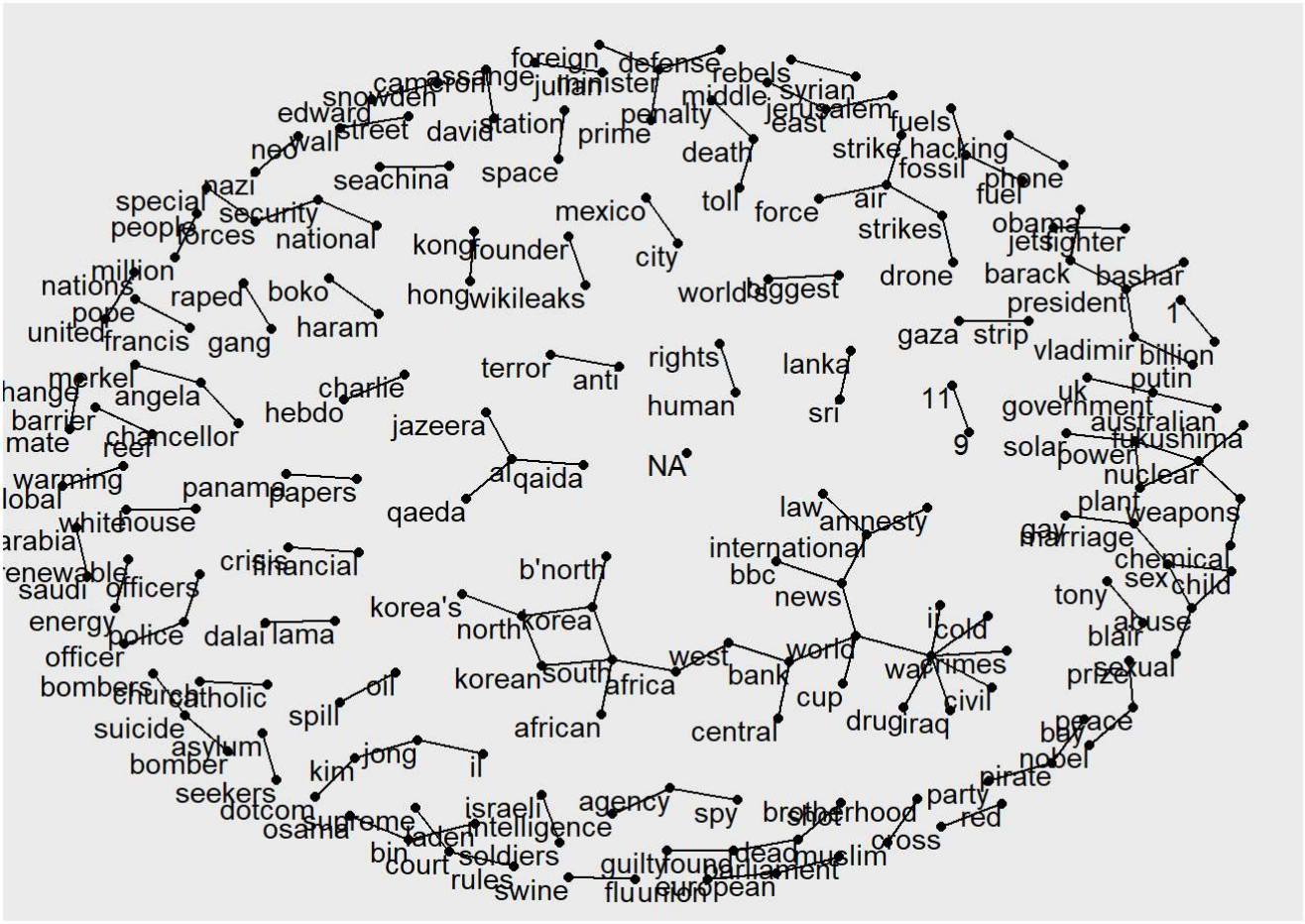
```
down_graph <- speech_bigrams %>%
count(word1, word2, sort=TRUE) %>%
filter(n > 20) %>%
graph_from_data_frame()
```

```
## Warning in graph_from_data_frame(.): In `d' `NA' elements were replaced
## with string "NA"
```

```
down_graph
```

```
## IGRAPH 4aff259 DN-- 195 132 --
## + attr: name (v/c), n (e/n)
## + edges from 4aff259 (vertex names):
##  [1] north   ->korea    prime   ->minister human   ->rights
##  [4] saudi   ->arabia   NA      ->NA       bbc     ->news
##  [7] climate ->change   west    ->bank     north   ->korean
## [10] south   ->korea    middle  ->east     hong    ->kong
## [13] al      ->qaeda    edward  ->snowden  kim     ->jong
## [16] war     ->crimes   vladimir->putin    julian  ->assange
## [19] south   ->africa   bin     ->laden    european->union
## [22] al      ->jazeera  pope    ->francis  united  ->nations
## + ... omitted several edges
```

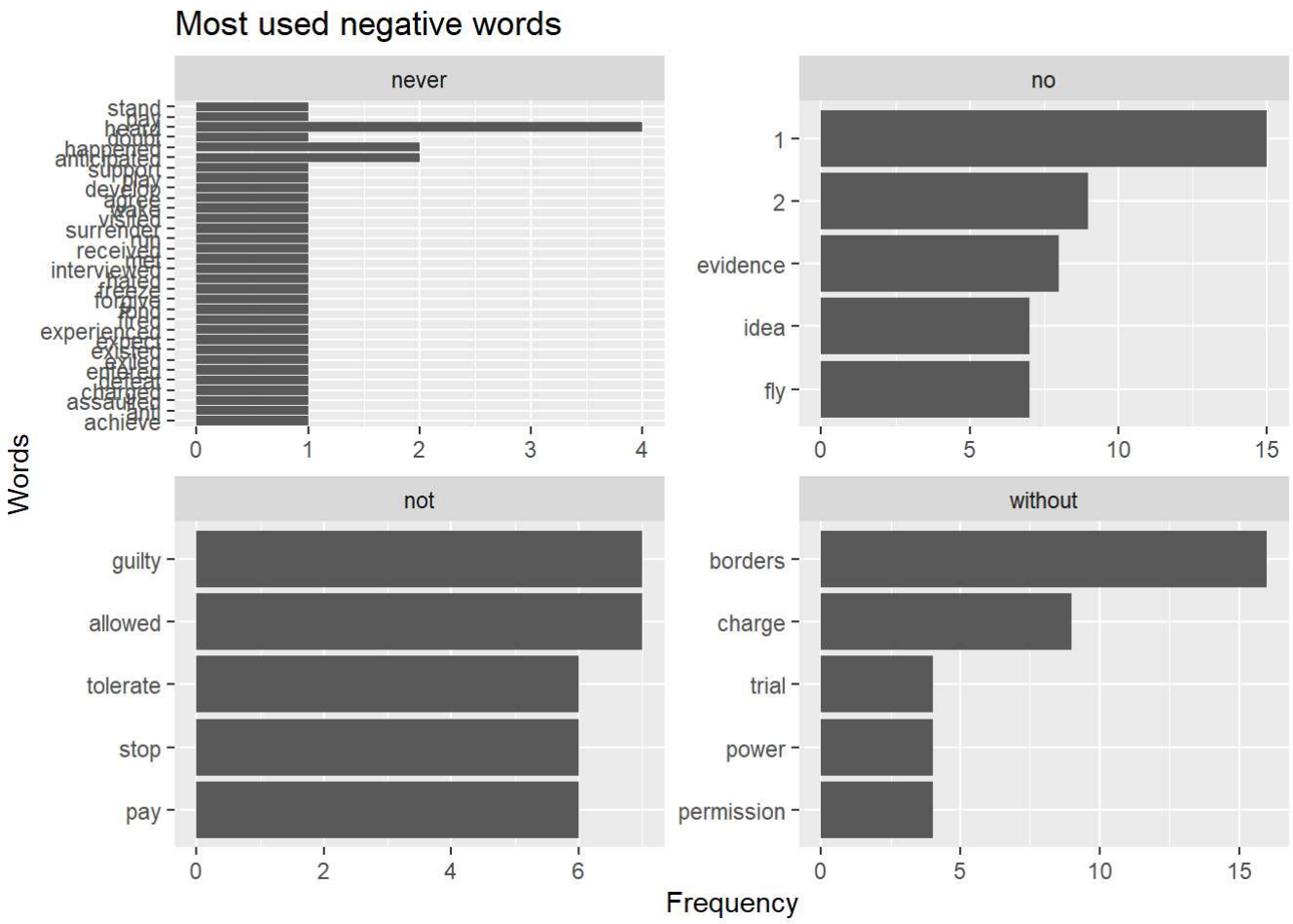
```
ggraph(down_graph,
layout="igraph",
algorithm="kk") +
geom_edge_link() +
geom_node_point() +
geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```



Most used negative words

```
speech_bigrams2 <- unnest_tokens(tspeech,
                                 bigram, text,
                                 token = "ngrams", n=2)

speech_bigrams2 <- speech_bigrams2 %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_bigrams2 <- speech_bigrams2 %>%
  filter(word1 %in% speech_negation$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word2 %in% speech_stop$word)

speech_bigrams2 %>%
  count(word1, word2, sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  mutate(word2 = reorder(word2, n)) %>%
  group_by(word1) %>%
  top_n(5)%>%
  ggplot(aes(word2, n)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~word1, scales="free") +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Most used negative words") +
  coord_flip()
```

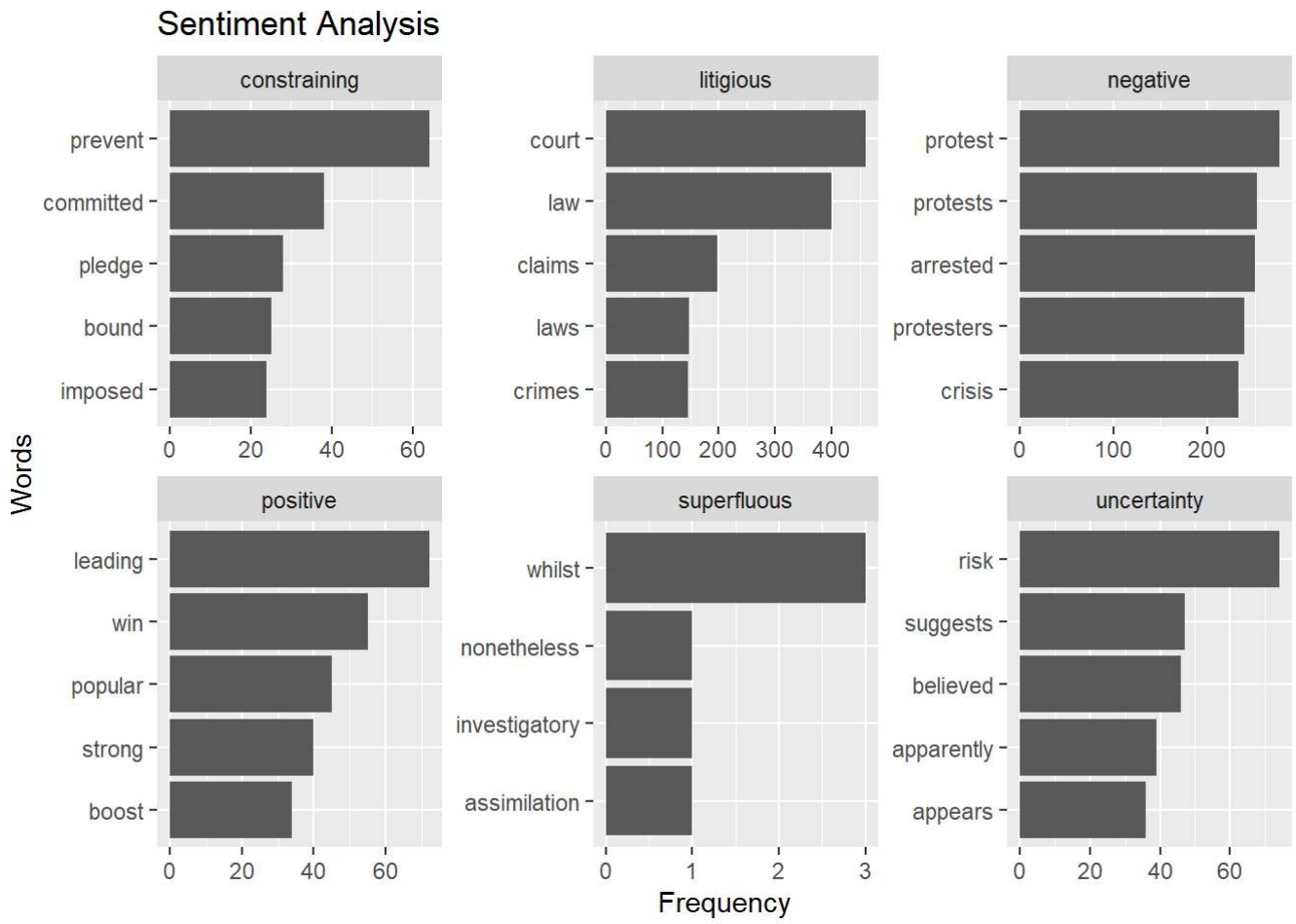
```
## Selecting by n
```



Sentiment Clustering and Analysis

```
speech_bigrams3 <- unnest_tokens(tspeech,
                                 bigram,
                                 text,
                                 token = "ngrams",
                                 n=2)

speech_bigrams3 <- speech_bigrams3 %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_bigrams3 <- speech_bigrams3 %>%
  filter(!word1 %in% speech_negation$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word2 %in% speech_stop$word)

loughlex <- get_sentiments("loughran")
speech_bigrams3 %>%
  inner_join(loughlex,
             by= c("word2"="word"))%>%
  count(sentiment,word2, sort=TRUE)%>%
  mutate(word = reorder(word2,n))%>%
  group_by(sentiment)%>%
  top_n(5)%>%
  ggplot(aes(x=word, y=n)) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~sentiment, scales = "free") +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Sentiment Analysis") +
  coord_flip()
```

```
## Selecting by word
```



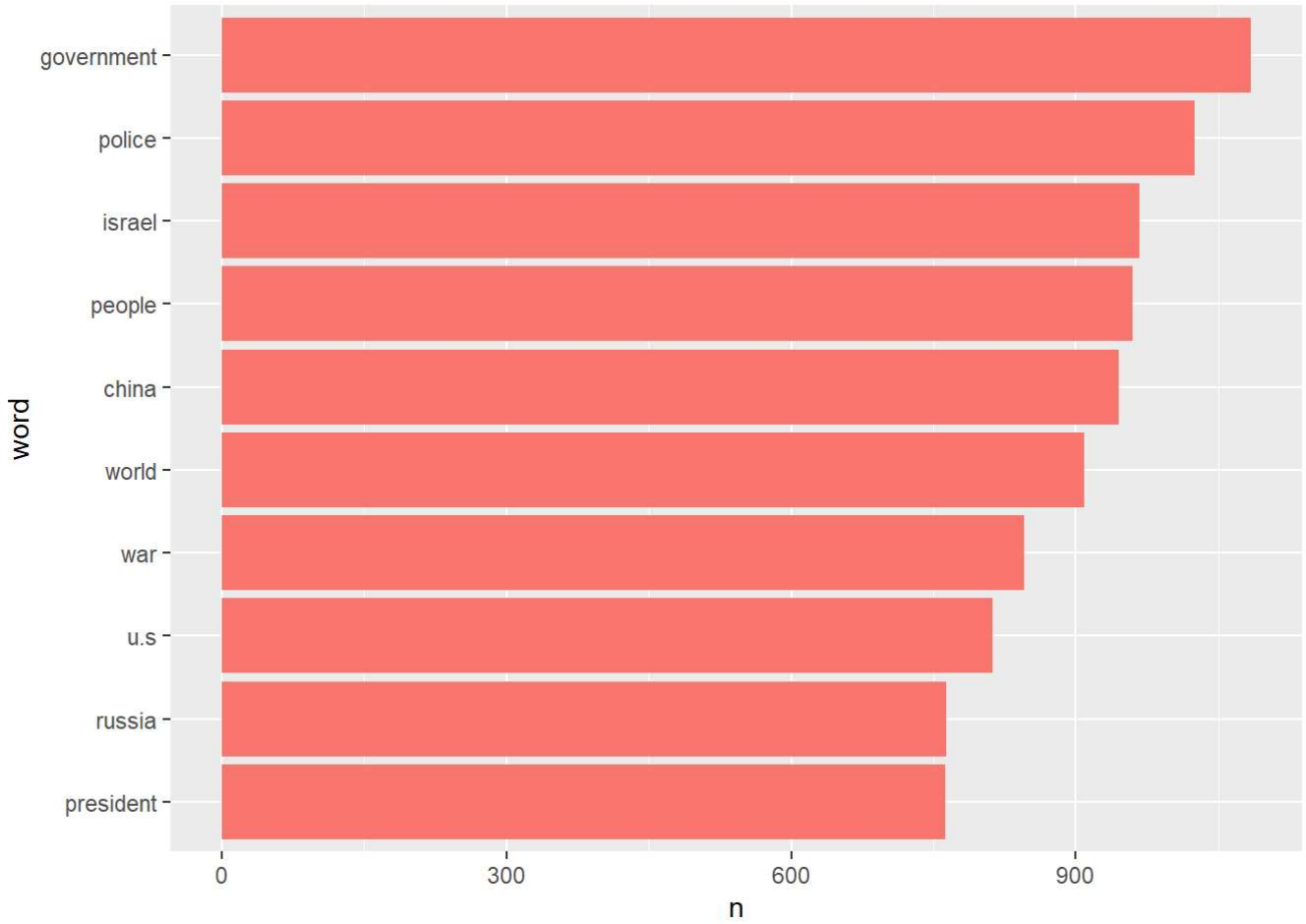
Up Market Analysis

```
speech <- read_lines("up_market.txt")

tspeech <- tibble(line=1:length(speech),text = speech)
```

Find Most Used Words

```
tspeech %>%
  unnest_tokens(word,text)%>%
anti_join(stop_words, by="word") %>%
count(word, sort=TRUE) %>%
filter(n > 150) %>%
mutate(word = reorder(word, n)) %>%
top_n(10)%>%
ggplot(aes(x=word, y=n, fill=rgb(0,1,0))) +
geom_col(show.legend = FALSE) +
coord_flip()
```

```
## Selecting by n
```



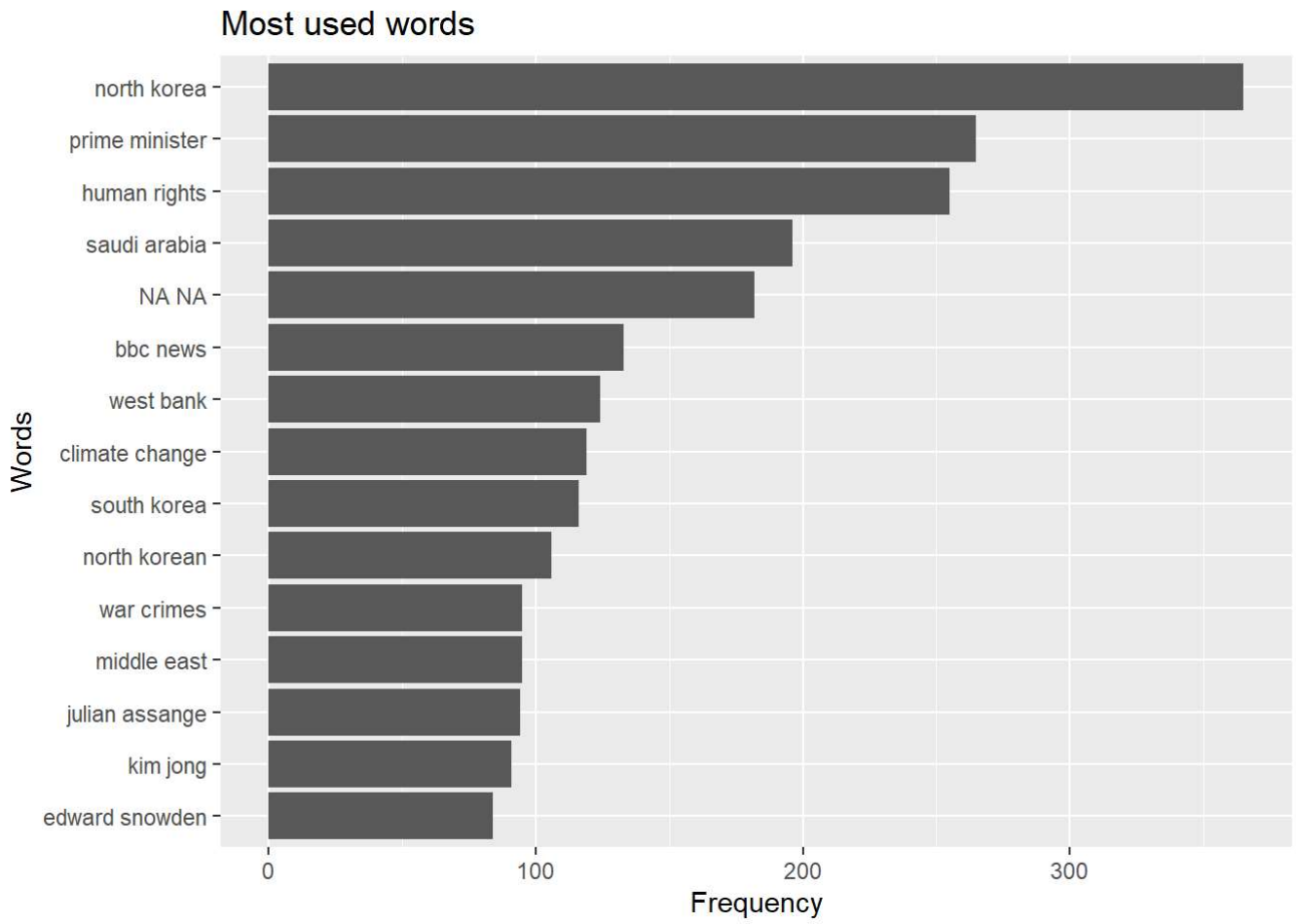
Most used Bigrams

```
tspeech <- tibble(line=1:length(speech),text = speech)

speech_bigrams <- unnest_tokens(tspeech, bigram, text,token = "ngrams", n=2)
speech_bigrams
```

```
## # A tibble: 474,190 x 2
##     line bigram
##    <int> <chr>
##  1     1 b'why wont
##  2     1 wont america
##  3     1 america and
##  4     1 and nato
##  5     1 nato help
##  6     1 help us
##  7     1 us if
##  8     1 if they
##  9     1 they wont
## 10     1 wont help
## # ... with 474,180 more rows
```

```
speech_bigrams <- speech_bigrams %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_stop <- tibble(word = c("applause"))



speech_bigrams <- speech_bigrams %>%
  filter(!word1 %in% stop_words$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word1 %in% speech_stop$word)%>%
  filter(!word2 %in% speech_stop$word)

speech_negation <- tibble(word = c("never",
                                   "no",
                                   "without",
                                   "not"))

speech_bigrams <- speech_bigrams %>%
  filter(!word1 %in% speech_negation$word)

speech_bigrams %>%
  count(word1,word2,sort=TRUE) %>%
  unite(bigram,c(word1,word2),sep=" ")%>%
  top_n(15) %>%
  mutate(word= reorder(bigram,n))%>%
  ggplot(aes(x=word,y=n)) +
  geom_col() +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Most used words") +
  coord_flip()
```

```
## Selecting by n
```



Bigram Connector graph

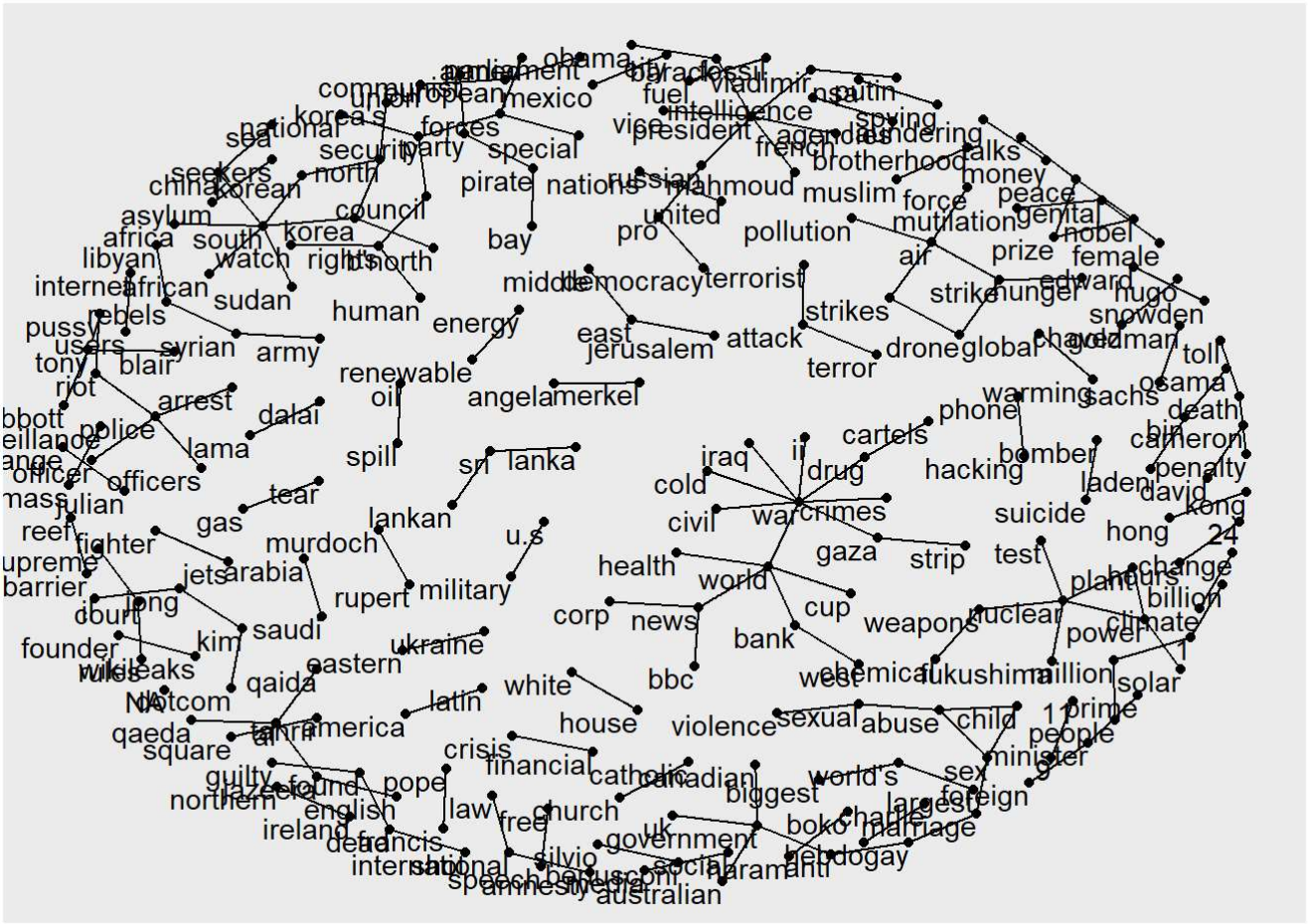
```
library(igraph)
library(ggraph)
up_graph <- speech_bigrams %>%
count(word1, word2, sort=TRUE) %>%
filter(n > 20) %>%
graph_from_data_frame()
```

```
## Warning in graph_from_data_frame(.): In `d' `NA' elements were replaced
## with string "NA"
```

```
up_graph
```

```
## IGRAPH 5231f51 DN-- 240 172 --
## + attr: name (v/c), n (e/n)
## + edges from 5231f51 (vertex names):
##  [1] north   ->korea    prime   ->minister human   ->rights
##  [4] saudi   ->arabia   NA      ->NA       bbc     ->news
##  [7] west    ->bank     climate ->change   south   ->korea
## [10] north   ->korean   middle  ->east     war     ->crimes
## [13] julian  ->assange  kim     ->jong     edward  ->snowden
## [16] al      ->jazeera  vladimir->putin    world   ->cup
## [19] united  ->nations  al      ->qaeda    hong    ->kong
## [22] security->forces   supreme ->court    david   ->cameron
## + ... omitted several edges
```

```
ggraph(up_graph,
layout="igraph",
algorithm="kk") +
geom_edge_link() +
geom_node_point() +
geom_node_text(aes(label = name), vjust = 1, hjust = 1)
```



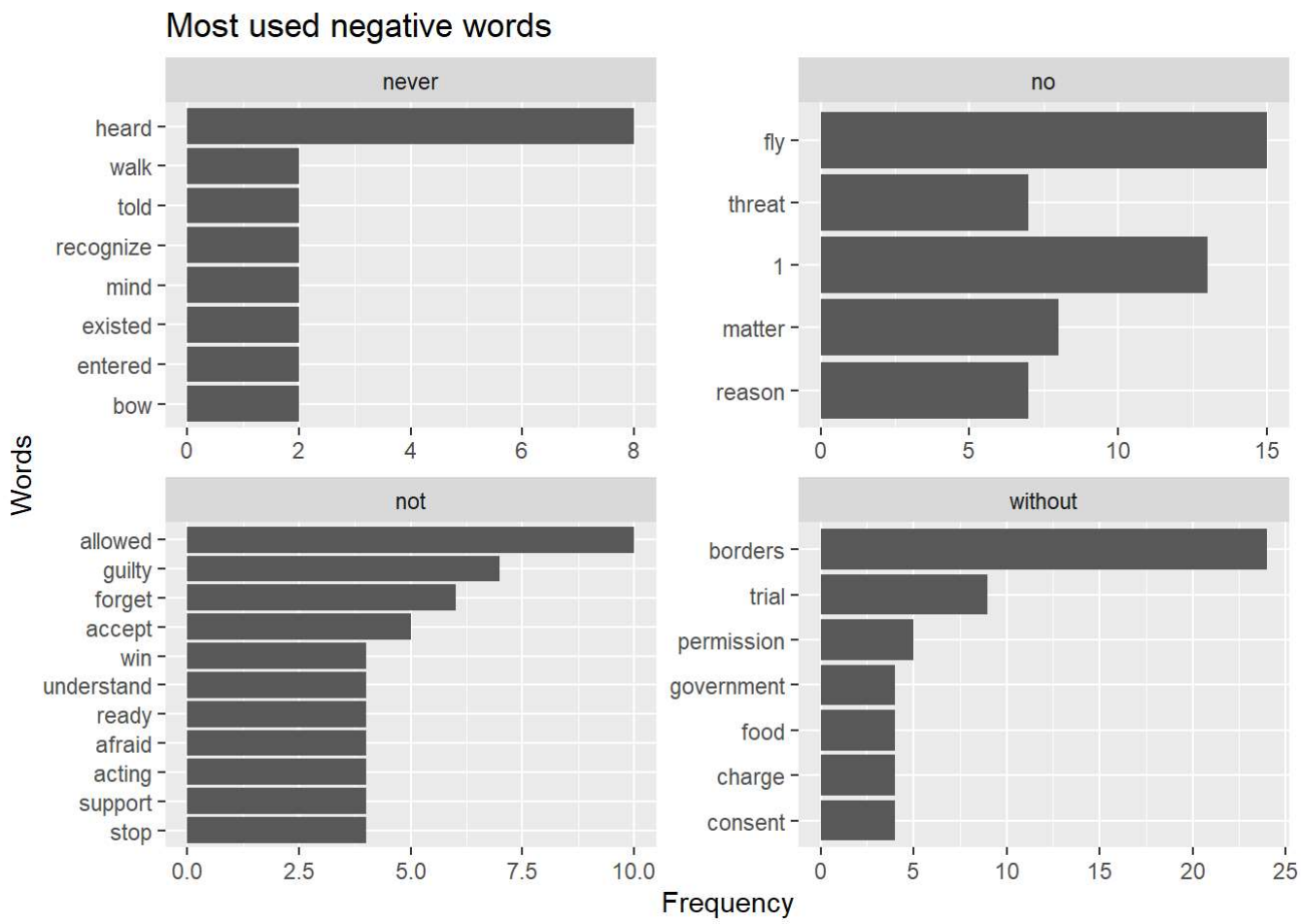
Most used negative words

```
speech_bigrams2 <- unnest_tokens(tspeech,
                                 bigram, text,
                                 token = "ngrams", n=2)
speech_bigrams2
```

```
## # A tibble: 474,190 x 2
##     line bigram
##    <int> <chr>
##  1     1 b'why wont
##  2     1 wont america
##  3     1 america and
##  4     1 and nato
##  5     1 nato help
##  6     1 help us
##  7     1 us if
##  8     1 if they
##  9     1 they wont
## 10     1 wont help
## # ... with 474,180 more rows
```

```
speech_bigrams2 <- speech_bigrams2 %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_bigrams2 <- speech_bigrams2 %>%
  filter(word1 %in% speech_negation$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word2 %in% speech_stop$word)

speech_bigrams2 %>%
  count(word1, word2, sort = TRUE) %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  mutate(word2 = reorder(word2, n)) %>%
  group_by(word1) %>%
  top_n(5)%>%
  ggplot(aes(word2, n)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~word1, scales="free") +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Most used negative words") +
  coord_flip()
```

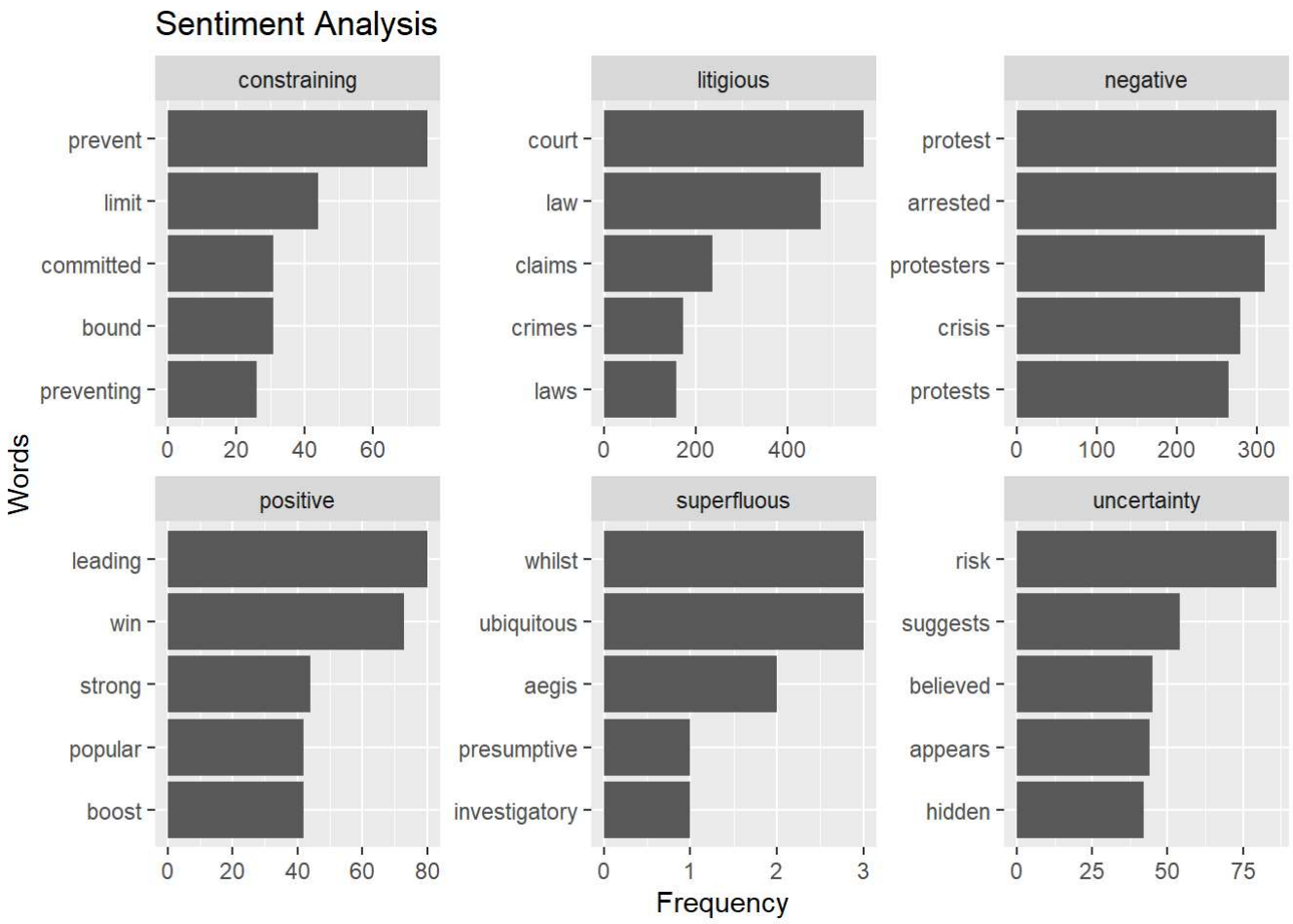
```
## Selecting by n
```



Sentiment Clustering and Analysis

```
speech_bigrams3 <- unnest_tokens(tspeech,
                                 bigram,
                                 text,
                                 token = "ngrams",
                                 n=2)

speech_bigrams3 <- speech_bigrams3 %>%
  separate(bigram, c("word1","word2"), sep=" ")

speech_bigrams3 <- speech_bigrams3 %>%
  filter(!word1 %in% speech_negation$word)%>%
  filter(!word2 %in% stop_words$word) %>%
  filter(!word2 %in% speech_stop$word)

loughlex <- get_sentiments("loughran")
speech_bigrams3 %>%
  inner_join(loughlex,
             by= c("word2"="word"))%>%
  count(sentiment,word2, sort=TRUE)%>%
  mutate(word = reorder(word2,n))%>%
  group_by(sentiment)%>%
  top_n(5)%>%
  ggplot(aes(x=word, y=n)) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~sentiment, scales = "free") +
  ylab("Frequency") +
  xlab("Words") +
  ggtitle("Sentiment Analysis") +
  coord_flip()
```

```
## Selecting by word
```


Sentiment Analysis
constraining
prevent
limit
committed
bound
preventing
0
20
40
60
litigious
court
law
claims
crimes
laws
0
200
400
negative
protest
arrested
protesters
crisis
protests
0
100
200
300
positive
leading
win
strong
popular
boost
0
20
40
60
80
superfluous
whilst
ubiquitous
aegis
presumptive
investigatory
0
1
2
3
uncertainty
risk
suggests
believed
appears
hidden
0
25
50
75
Words
Frequency