Data 607 Week 10 Assignment

Jean Jimenez, Matthew Roland, Kelly Eng

2023-10-30

Introduction

For this weeks homework assignment, we had to preform sentiment analysis on some text. Sentiment analysis is when you analyze some text, and attribute values to the word used. It is used to figure out opinions or sentiment held in writing.

There are many different Lexicons that can be used to conduct sentiment analysis. For the purpose of this assignment, we are using `nrc` and `loughran`. There are many different uses of sentiment analysis.

In this assignment, we decided to conduct sentiment analysis on New York Time's articles about Israeli Prime Minister Benjamin Netanyahu. The ongoing conflict has caused polarization in media and we were interested to see how the NYT's articles about Mr.Netanyahu rate in sentiment. We believe that sentiment analysis can be a useful tool to combat biases and misinformation in media/ in the news.

Disclaimer: The sentiment analysis conducted in this assignment is a **PURLEY** academic exercise aimed at understanding and applying data science techniques. It is **NOT** intended to promote, endorse, or engage in any form of criticism against any individual or government, including the Israeli government. This analysis should **NOT** be misconstrued or misinterpreted as a political statement or an act of anti-semitism. As data science students and American citizens, we exercise our protected right to conduct and discuss factual, evidence-based research on public figures and world leaders in accordance with ethical and moral standards. Our commitment as data scientists is to report accurately, use truthful data, and avoid misleading representations in all our work.

Work

Obtaining Data

To obtained the data, we used the NYT's article search API. We asked the API to return articles about Mr.Netanyahu for the past 3 months. Afterwards, we filtered down the top 5 articles and exported into a csv (because the API search results will change over time).

```r
#loading libraries
library(httr)
library(jsonlite)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(tidytext)
library(textdata)
```

```
##
## Attaching package: 'textdata'
##
## The following object is masked from 'package:httr':
##
##     cache_info
```

```
library(cleanNLP)
library(ggplot2)
library(knitr)
```

The snippet and lead paragraph section of the MetaData is not enough data to conduct the sentiment analysis. However, I have a NYT subscription. For the first 5 articles, I obtained the full text of the articles and uploaded them to github as a `txt` file. Then, I loaded all of the full texts from Github.

```
#CSV from above
csv_link="https://raw.githubusercontent.com/sleepysloth12/data607_wk10/main/netan_dat.csv"

netan_dat=read.csv(url(csv_link))

#prefix for github full txt folder
full_txt_raw_url_prefix="https://raw.githubusercontent.com/sleepysloth12/data607_wk10/main/fullText/"

#full txt url column with link to full txt.
netan_dat= netan_dat %>%
  filter(article_text_file != "") %>%
  mutate(full_txt_url= paste0(full_txt_raw_url_prefix,article_text_file,".txt") )

#All full texts are in this data frame in the full_text column
netan_dat_txt = netan_dat %>%
  rowwise() %>%
  mutate(full_txt = list(readLines(full_txt_url))) %>%
  ungroup()
```

```
## Warning: There were 4 warnings in `mutate()`.
## The first warning was:
## i In argument: `full_txt = list(readLines(full_txt_url))`.
## i In row 2.
## Caused by warning in `readLines()`:
## ! incomplete final line found on 'https://raw.githubusercontent.com/sleepysloth12/data607_wk10/main/
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

Tokenization & Sentiment Analysis

The first step of performing sentiment analysis is tokenization. Tokenization is dividing the text into distinct parts and attributing a value to that token.

We added stop words to each of the articles to remove words that do not necessarily influence the sentiment of the article.

We used the `nrc` and `loughran` lexicon.

No terms or conditions appeared when downloading the `nrc` lexicon. A license is not needed for academic use for the `loughran` lexicon.

First Article

The first article was Benjamin Netanyahu's Two Decades of Power, Bluster and Ego, published to the NY Time's Magazine section on September 27th, 2023 and updated on October 6th, 2023. This article has 8222 words and we decided to tokenize it by paragraph.

```
#First Article

first_txt= netan_dat_txt$full_txt[1]
class(first_txt)
```

```
## [1] "list"
```

```
first_txt=first_txt[[1]]
tot_first_len=length(first_txt)

#getting each paragraph
first_article_paragraphs=c()


for (i in 1:tot_first_len){
  if(first_txt[i] != ""){
    first_article_paragraphs=c(first_article_paragraphs,first_txt[i])}
}

#all Paragraphs of the first article
#first_article_paragraphs


first_df = tibble(paragraph = 1:length(first_article_paragraphs), text = first_article_paragraphs)

# Tidy article split apart by paragraphs
tidy_art1 <- first_df %>%
  group_by(paragraph) %>%
  ungroup() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```
nrc_sentiments=get_sentiments("nrc")


#obj_1 = cnlp_annotate(first_df$text)

#NRC Results first article
sar_1_nrc = first_df %>%
  unnest_tokens(word, text) %>%
  inner_join(nrc_sentiments, by = "word") %>%
  group_by(paragraph, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
```

```r
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relationsh
## i Row 4 of `x` matches multiple rows in `y`.
## i Row 9068 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.

## `summarise()` has grouped output by 'paragraph'. You can override using the
## `.groups` argument.
```

```r
nrc_pos_and_neg <- nrc_sentiments %>%
  filter(sentiment %in% c("positive", "negative"))

##Certain neutral terms such as public, prime, government, foreign, vote, president, government, etc. w

custom_stop <- bind_rows(tibble(word = c("deal", "public", "prime", "president", "including", "governmen
                                lexicon = c("custom")),
                         stop_words)

nrc_pos_and_neg <- nrc_pos_and_neg %>% anti_join(custom_stop)
```

```
## Joining with `by = join_by(word)`
```

```r
custom_stop
```

```
## # A tibble: 1,161 x 2
##    word       lexicon
##    <chr>      <chr>
##  1 deal       custom
##  2 public     custom
##  3 prime      custom
##  4 president  custom
##  5 including  custom
##  6 government custom
##  7 foreign    custom
##  8 vote       custom
##  9 serve      custom
## 10 john       custom
## # i 1,151 more rows
```

```r
# The additional lexicon sentiment not shown in tidytextmining.com
# It's from the the textdata package
loughran <- get_sentiments("loughran")

loughran_pos_and_neg <- loughran %>%
  filter(sentiment %in% c("positive", "negative"))

sar_1_loughran <- tidy_art1 %>%
  inner_join(loughran, by = "word") %>%
  group_by(paragraph, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., loughran, by = "word"): Detected an unexpected many-to-many relationship bet
## i Row 405 of `x` matches multiple rows in `y`.
## i Row 1773 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'paragraph'. You can override using the
## `.groups` argument.
```

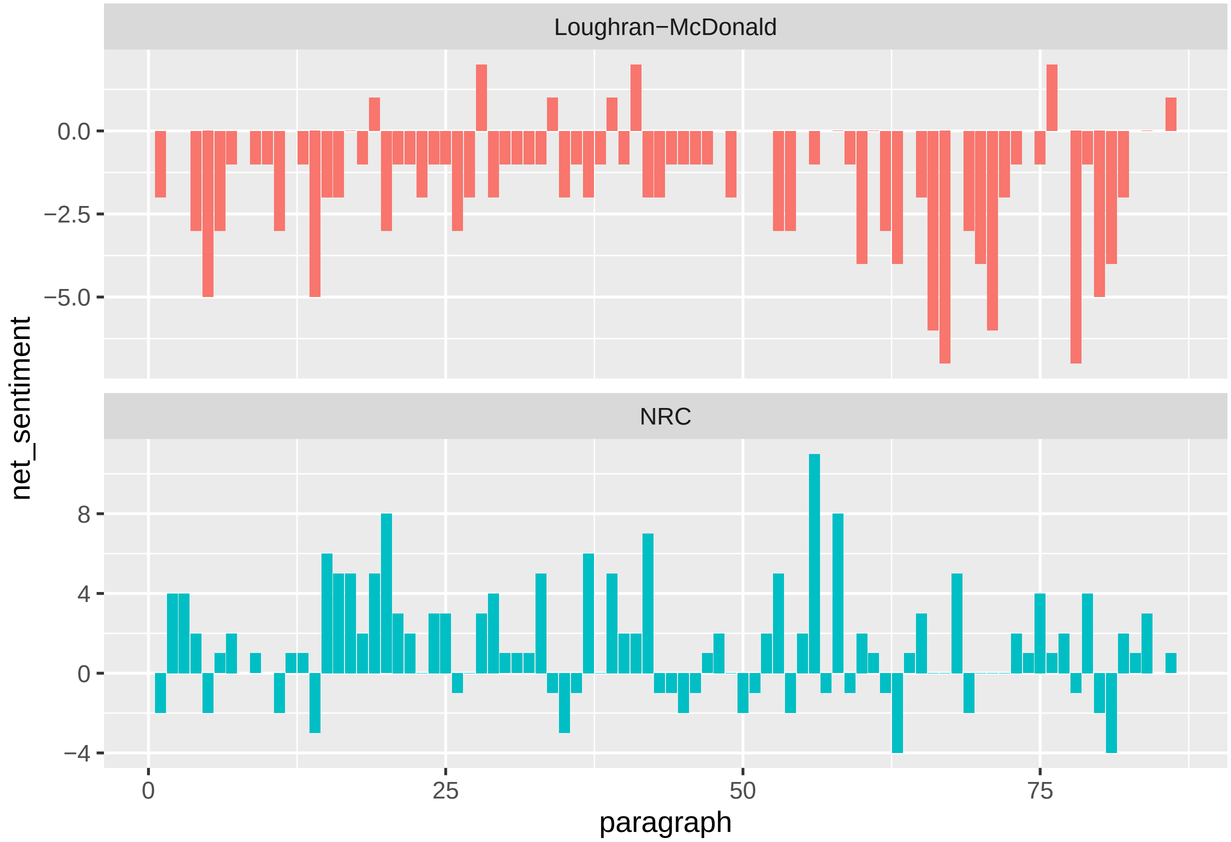
```
nrc_and_loughran_art1 <- bind_rows(
  tidy_art1 %>%
    inner_join(nrc_pos_and_neg) %>%
    mutate(method = "NRC"),
  tidy_art1 %>%
    inner_join(loughran_pos_and_neg) %>%
    mutate(method = "Loughran-McDonald")) %>%
  group_by(paragraph, sentiment, method) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., nrc_pos_and_neg): Detected an unexpected many-to-many relationship between
## i Row 61 of `x` matches multiple rows in `y`.
## i Row 876 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'paragraph', 'sentiment'. You can override
## using the `.groups` argument.
```

```
# Comparison between the NRC & Loughran sentiment lexicons for article 1
nrc_and_loughran_art1 %>%
  ggplot(aes(paragraph, net_sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```

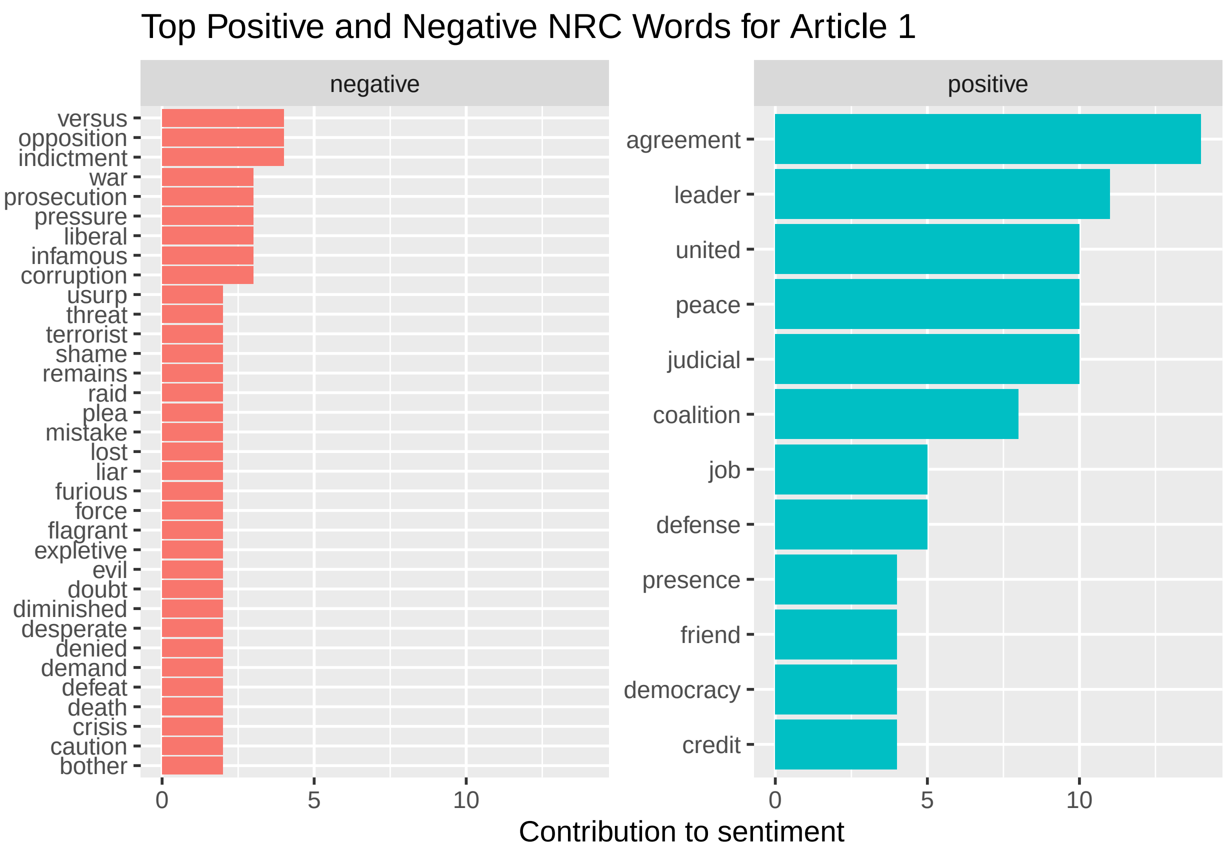


```r
art1_nrc_word_counts <- tidy_art1 %>%
  inner_join(nrc_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., nrc_pos_and_neg): Detected an unexpected many-to-many relationship between
## i Row 61 of `x` matches multiple rows in `y`.
## i Row 876 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

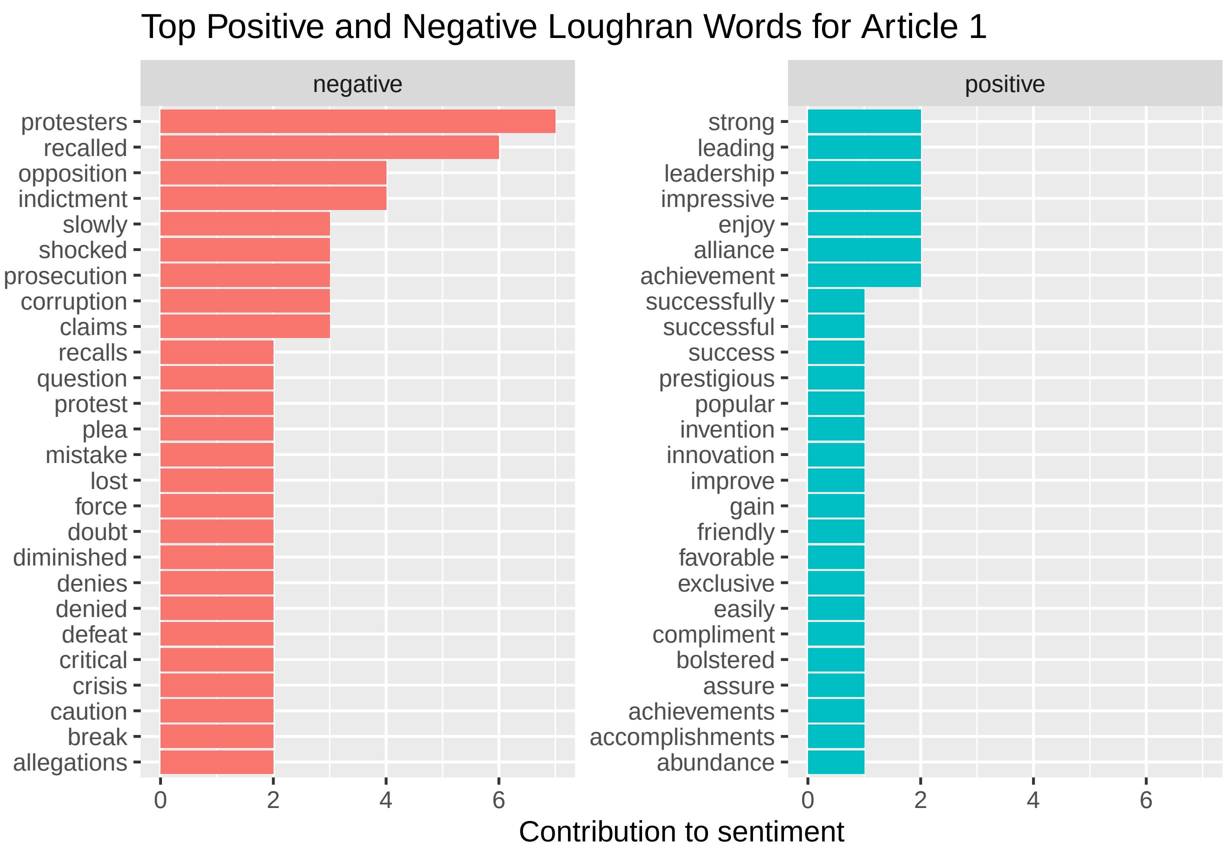
```r
art1_nrc_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative NRC Words for Article 1")
```



```
art1_loughran_word_counts <- tidy_art1 %>%
  inner_join(loughran_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
art1_loughran_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) +
  ggtitle("Top Positive and Negative Loughran Words for Article 1")
```



This article was published prior to the October 7th attacks on Israel. Prior to the attacks, the Israeli public has been critical of Mr. Netanyahu's rule. There have been protests leading up to this moment. Netanyahu, facing corruption charges, teamed up with a right-wing coalition to obtain majority and remain prime minister.

The `nrc` sentiment analysis picked up more positive sentiment for the article, while the `loughram` picked up more negative terms. In the context of the article, I think that the `Loughran` lexicon did a better job picking up the sentimemnt. This article provides a long history of Mr.Netanyahu's rise to power. The most common negative words that influenced this sentiment were 'protesters', 'recalled', 'opposition'.

Second Article

The second article was Netanyahu Must Go, published to the Opinion section of the New York Times on October 25th, 2023. The article is a transcript of an opinion short by an Israeli journalist and historian. The transcript has 169 words. For this text, we decided to separate by sentence.

```
second_txt= netan_dat_txt$full_txt[2]
class(second_txt)
```

```
## [1] "list"
```

```
second_txt=second_txt[[1]]
tot_second_len=length(second_txt)

#getting each sentence
second_article_sentences=c()
```

```r
for (i in 1:tot_second_len){
  if(second_txt[i] != ""){
    second_article_sentences=c(second_article_sentences,second_txt[i])}
}

#all Paragraphs of the 2nd article
#second_article_sentences

second_article_sentences = lapply(second_article_sentences, function(x) unlist(str_split(x, "\\. ")))
second_article_sentences = unlist(second_article_sentences)


second_df = tibble(sentence = 1:length(second_article_sentences), text = second_article_sentences)




#obj_2 = cnlp_annotate(second_df$text)

tidy_art2 <- second_df %>%
  group_by(sentence) %>%
  ungroup() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```r
custom_stop <- custom_stop %>% bind_rows(tibble(word = c("question", "quote", "mediterranean", "choice"
                                lexicon = c("custom")),
                         stop_words)

tidy_art2 <- tidy_art2 %>% anti_join(custom_stop)
```

```
## Joining with `by = join_by(word)`
```

```r
#NRC Results second article
sar_2_nrc = second_df %>%
  unnest_tokens(word, text) %>%
  inner_join(nrc_sentiments, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relationsh
## i Row 6 of `x` matches multiple rows in `y`.
## i Row 9977 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

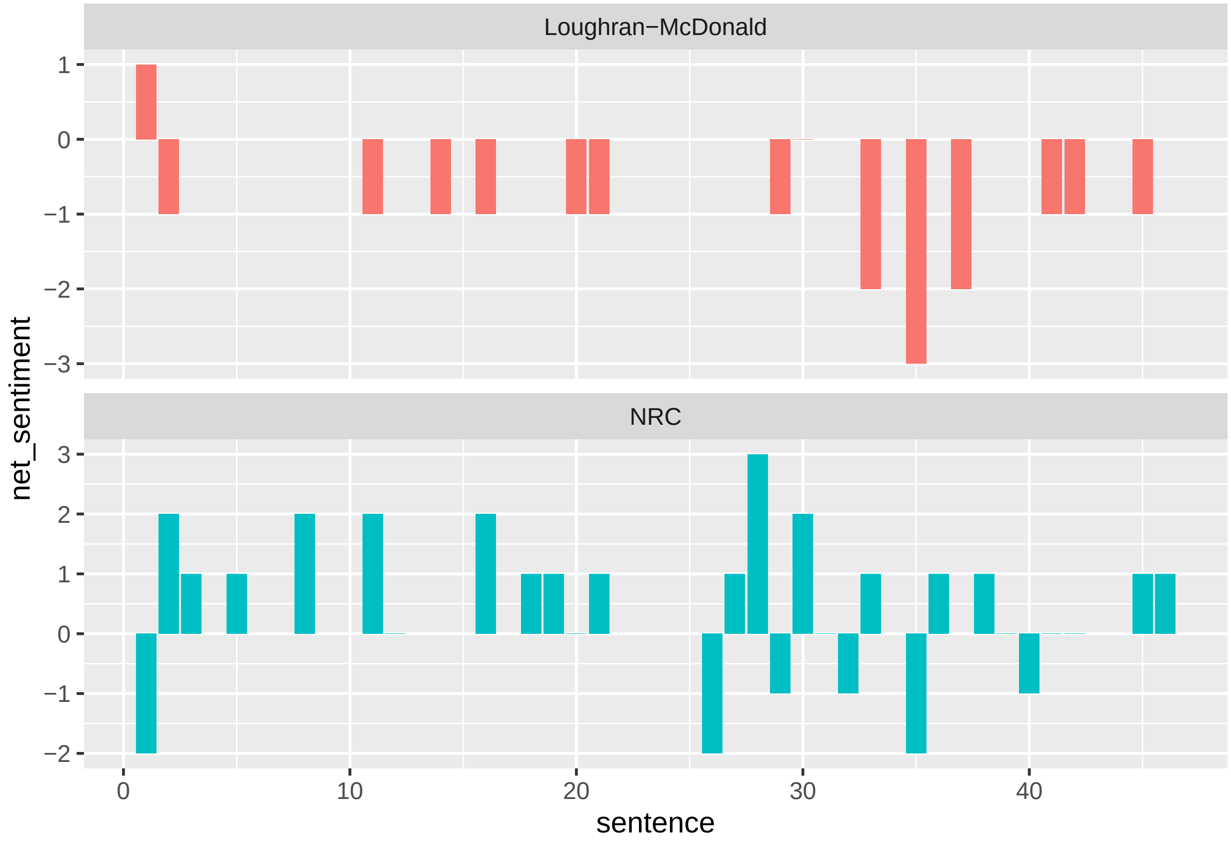
```r
#Loughran Results second article
sar_2_loughran <- tidy_art2 %>%
  inner_join(loughran, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

```
nrc_and_loughran_art2 <- bind_rows(
  tidy_art2 %>%
    inner_join(nrc_pos_and_neg) %>%
    mutate(method = "NRC"),
  tidy_art2 %>%
    inner_join(loughran_pos_and_neg) %>%
    mutate(method = "Loughran-McDonald")) %>%
  group_by(sentence, sentiment, method) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Joining with `by = join_by(word)`
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'sentence', 'sentiment'. You can override
## using the `.groups` argument.
```

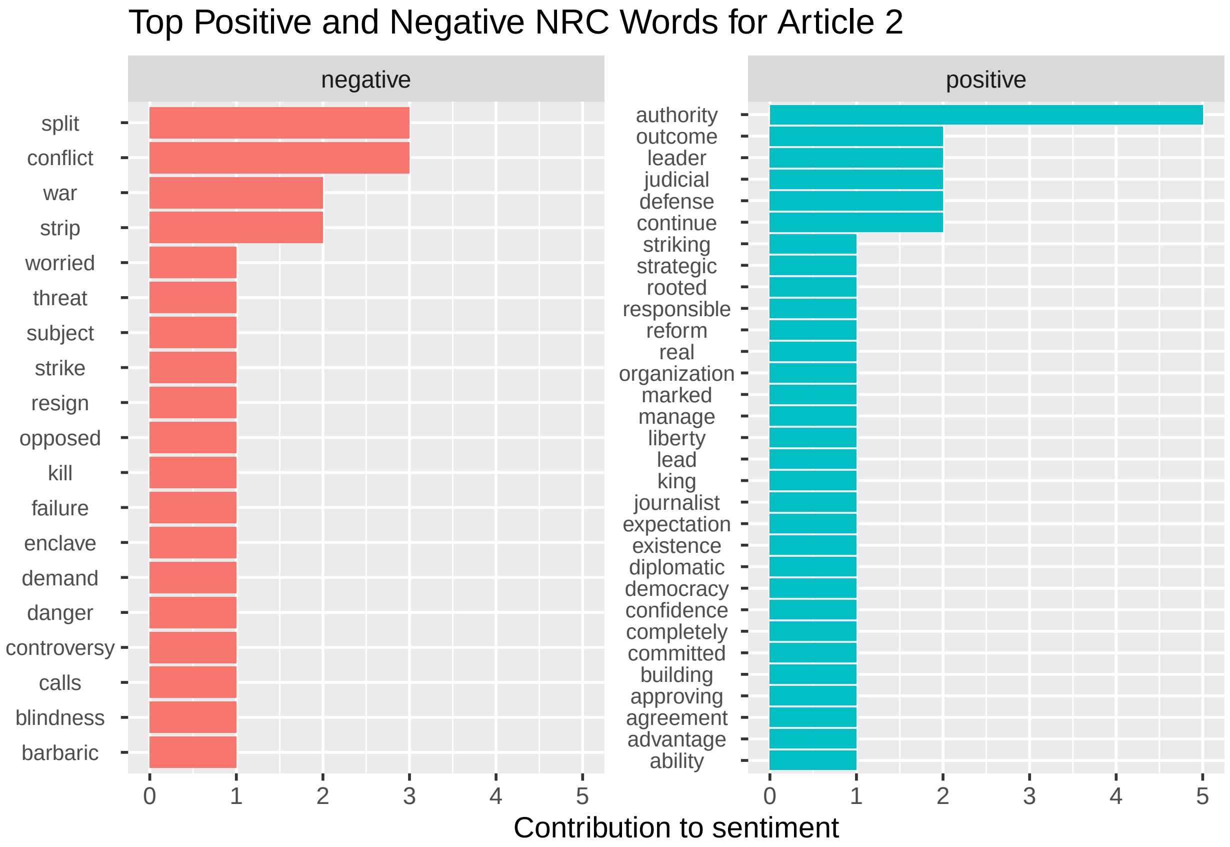
```
# Comparison between the NRC & Loughran sentiment lexicons for article 2
nrc_and_loughran_art2 %>%
  ggplot(aes(sentence, net_sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



```
art2_nrc_word_counts <- tidy_art2 %>%
  inner_join(nrc_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

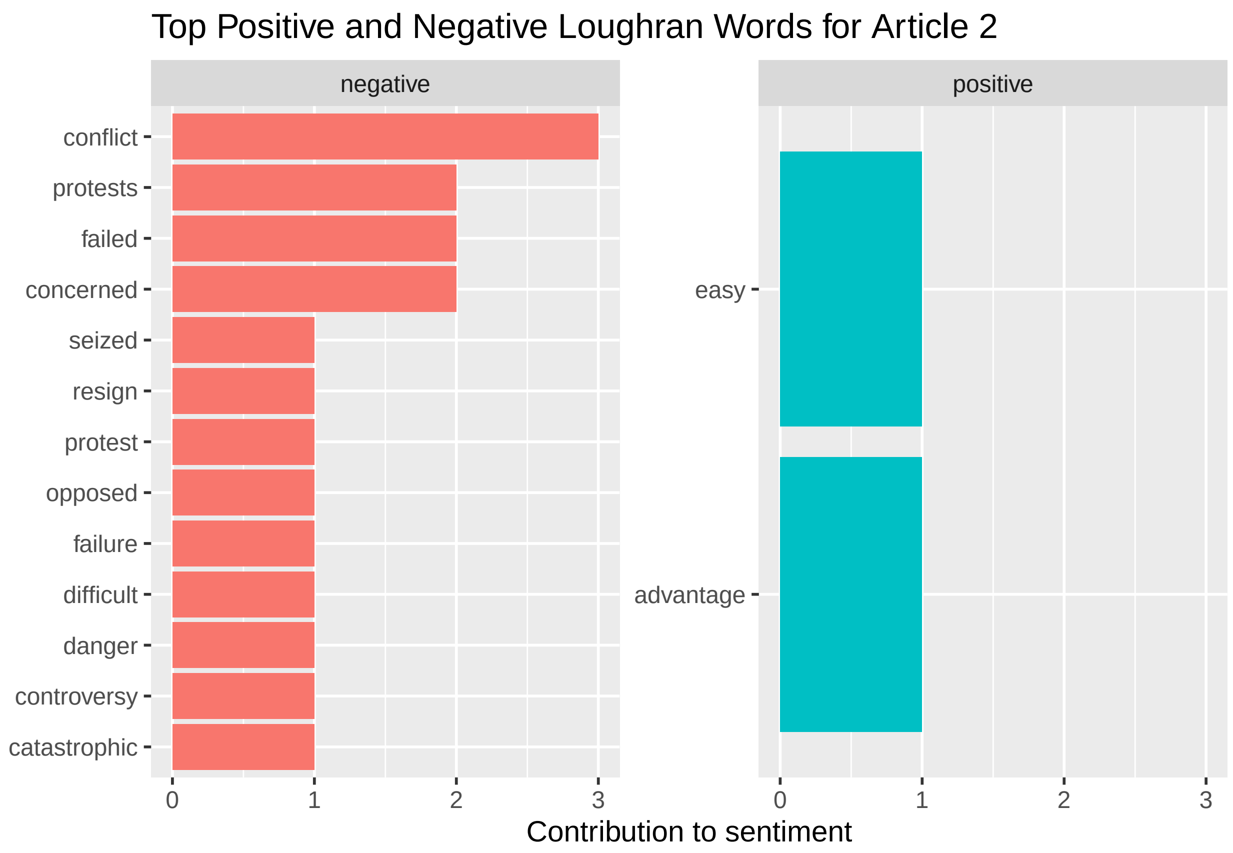
```
art2_nrc_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative NRC Words for Article 2") +
  theme(axis.text.y = element_text(angle = 0, hjust = .5, size = 8))
```



```
art2_loughran_word_counts <- tidy_art2 %>%
  inner_join(loughran_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
art2_loughran_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) +
  ggtitle("Top Positive and Negative Loughran Words for Article 2")
```



For this article, the `loughran` had a better sentiment analysis than the `nrc` lexicon. Some of the positive words picked up by the `nrc` such as "authority" and "striking" can be negative depending on the context. The `loughran` lexicon correctly identified words of negative sentiment, with "conflict", "protests", and "failed" being the most frequent.

Third Article

The third article was Netanyahu Rejects Calls for a Cease-fire — and for His Resignation. The article was published on October 30th, 2023 and has 335 words. It was published to the world news/middle east section. We separated this article by sentence.

```
third_txt= netan_dat_txt$full_txt[3]
class(third_txt)
```

```
## [1] "list"
```

```
third_txt=third_txt[[1]]
tot_third_len=length(third_txt)

#getting each sentence
third_article_sentences=c()


for (i in 1:tot_third_len){
  if(third_txt[i] != ""){
    third_article_sentences=c(third_article_sentences,third_txt[i])}
}
```

```
#all Paragraphs of the 3rd article
third_article_sentences
```

```
## [1] "Striking a defiant tone at a rare news briefing on Monday evening, Prime Minister Benjamin Neta
## [2] "Mr. Netanyahu’s political opponents have called for him to resign over his failure to stop the a
## [3] "Abroad, the conduct of the Israeli counterattack on Gaza - which has killed more than 8,000 peop
## [4] "Speaking to reporters in Tel Aviv, Mr. Netanyahu said that Israel would not agree to a halt in a
## [5] ""Just as the United States would not agree to a cease-fire after the bombing of Pearl Harbor or
## [6] "He then dismissed accusations that Israel is collectively punishing more than two million Gazans
## [7] "On Sunday, António Guterres, the U.N. secretary general, said the number of civilians killed in
## [8] "But Mr. Netanyahu said Israel was doing what it could to save civilian lives. He cited Israel’s
## [9] ""We’re going out of our way to prevent civilian casualties," Mr. Netanyahu said."
```

```
third_article_sentences = lapply(third_article_sentences, function(x) unlist(str_split(x, "\\. ")))
third_article_sentences = unlist(third_article_sentences)


third_df = tibble(sentence = 1:length(third_article_sentences), text = third_article_sentences)

#obj_3 = cnlp_annotate(third_df$text)

tidy_art3 <- third_df %>%
  group_by(sentence) %>%
  ungroup() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```
#NRC Results 3rd article
sar_3_nrc = third_df %>%
  unnest_tokens(word, text) %>%
  inner_join(nrc_sentiments, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relations
## i Row 3 of `x` matches multiple rows in `y`.
## i Row 5045 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

```
# Loughran results for 3rd article
sar_3_loughran <- tidy_art3 %>%
  inner_join(loughran, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

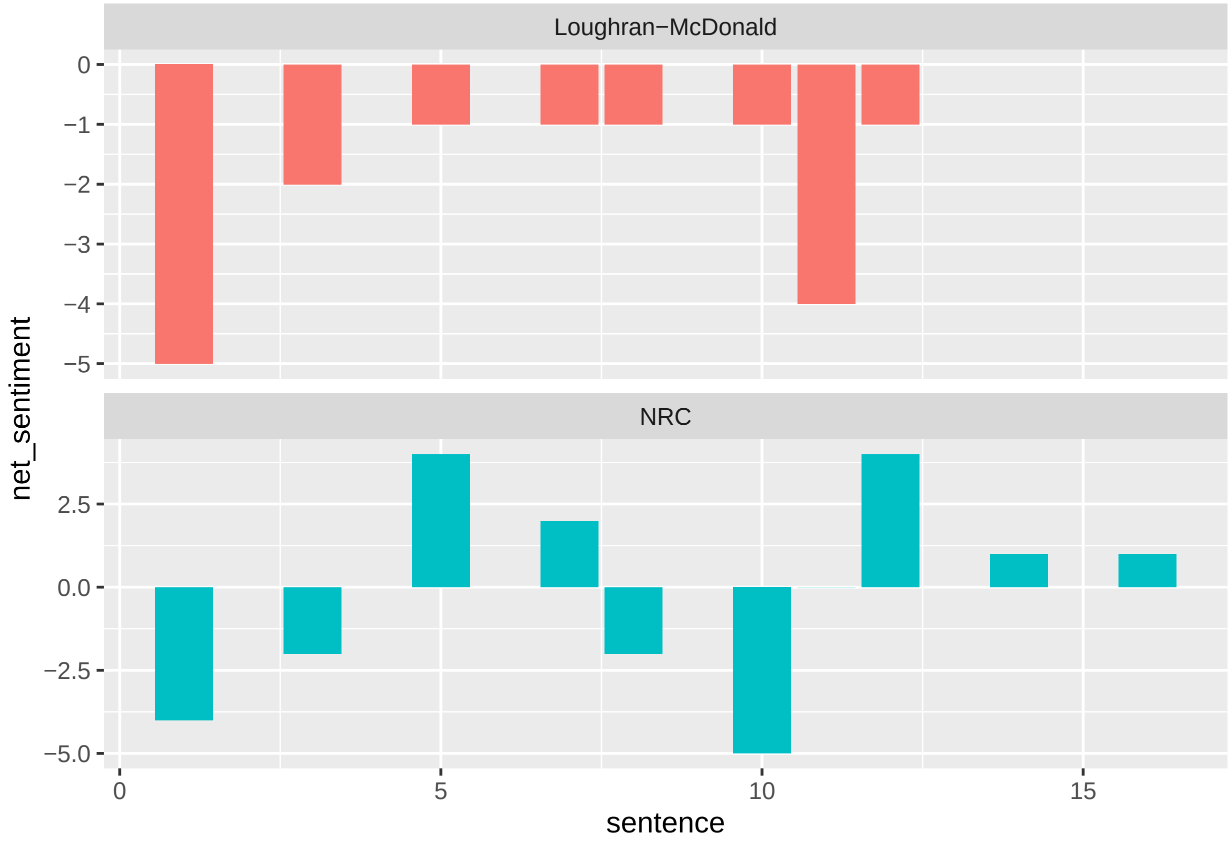
```
## Warning in inner_join(., loughran, by = "word"): Detected an unexpected many-to-many relationship be
## i Row 123 of `x` matches multiple rows in `y`.
## i Row 208 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

```
nrc_and_loughran_art3 <- bind_rows(
  tidy_art3 %>%
    inner_join(nrc_pos_and_neg) %>%
    mutate(method = "NRC"),
  tidy_art3 %>%
    inner_join(loughran_pos_and_neg) %>%
    mutate(method = "Loughran-McDonald")) %>%
  group_by(sentence, sentiment, method) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Joining with `by = join_by(word)`
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'sentence', 'sentiment'. You can override
## using the `.groups` argument.
```

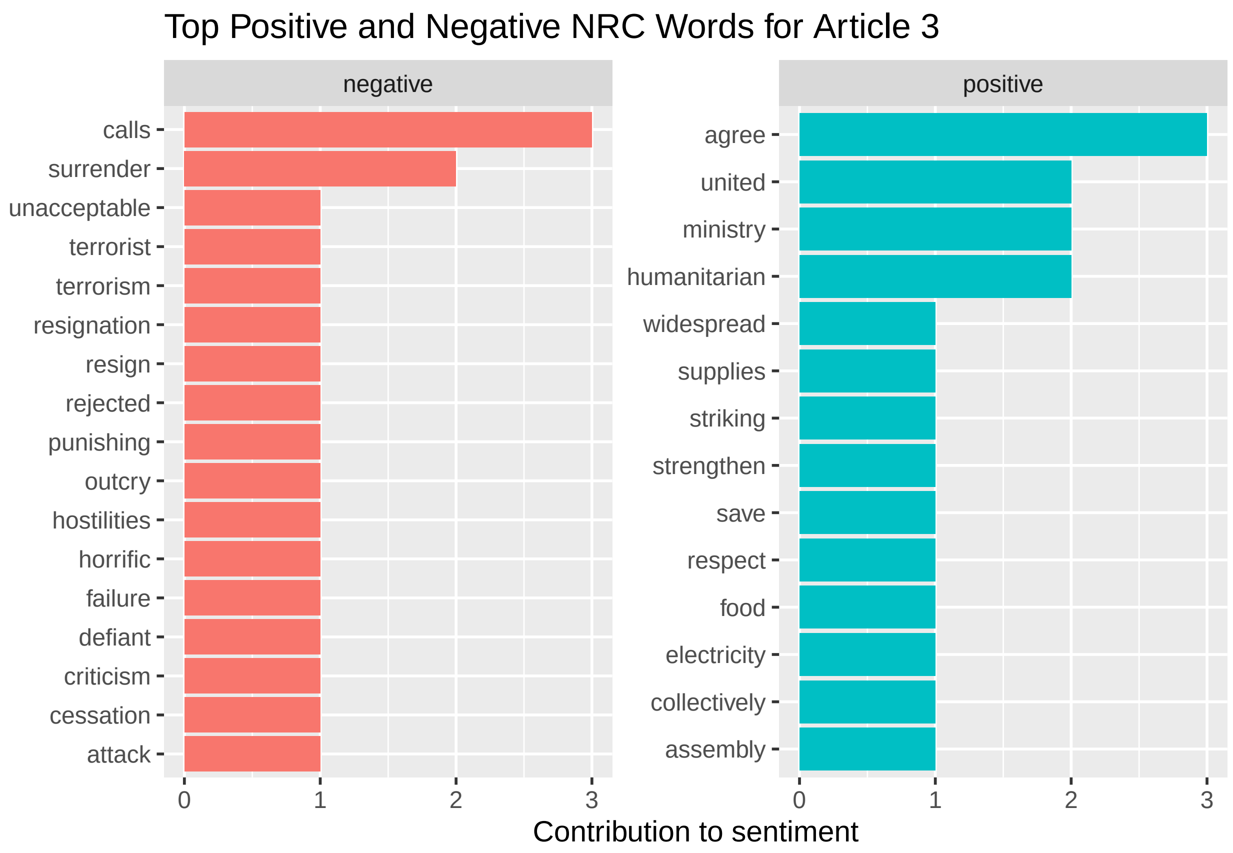
```
# Comparison between the NRC & Loughran sentiment lexicons for article 3
nrc_and_loughran_art3 %>%
  ggplot(aes(sentence, net_sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



```
art3_nrc_word_counts <- tidy_art3 %>%
  inner_join(nrc_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

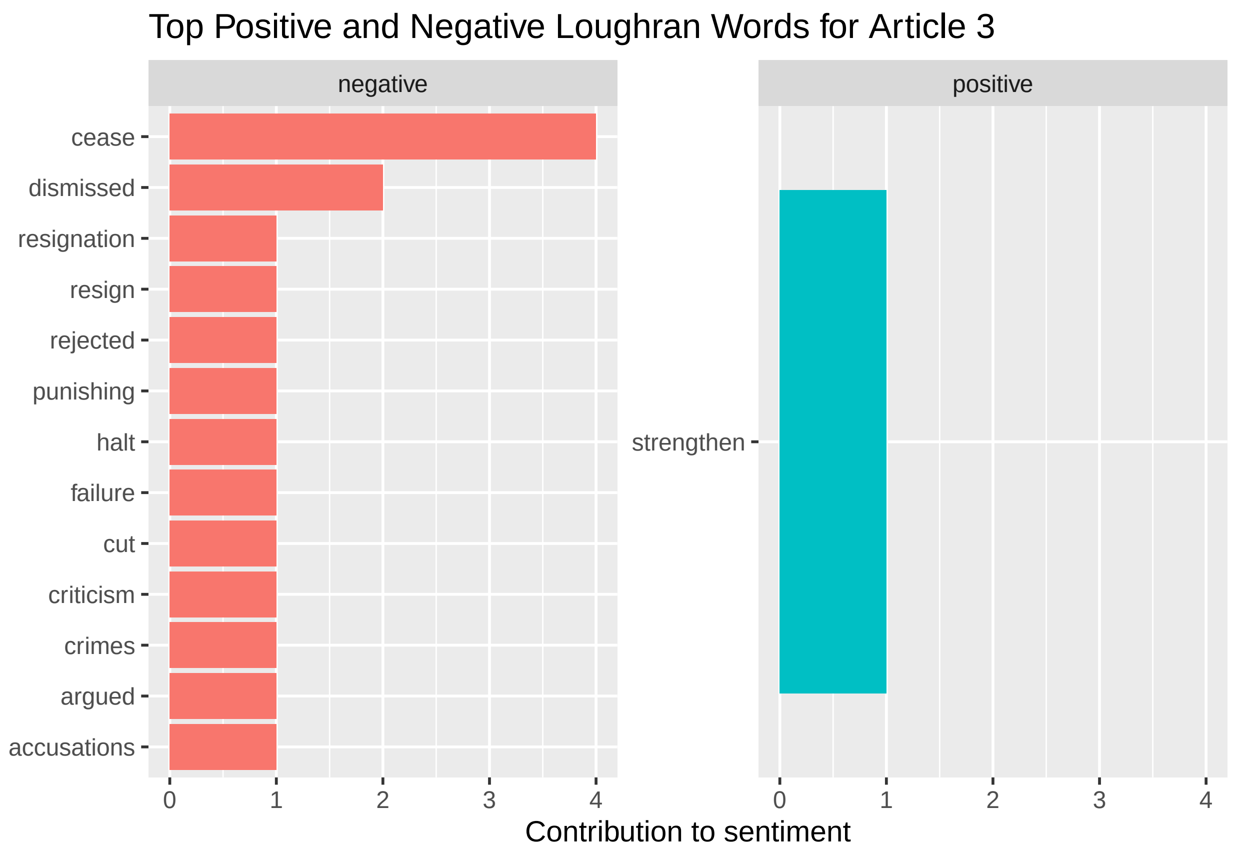
```
art3_nrc_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative NRC Words for Article 3")
```



```
art3_loughran_word_counts <- tidy_art3 %>%
  inner_join(loughran_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
art3_loughran_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) +
  ggtitle("Top Positive and Negative Loughran Words for Article 3")
```



The `loughran` lexicon did a better job than the `nrc` lexicon when it came to this article. The `nrc` lexicon incorrectly identified words such as "humanitarian" and "ministry" as positive sentiment. Meanwhile, in the context of the article, "humanitarian" has a negative connotation since it is talking about humanitarian workers criticism of the actions of Mr. Netanyahu's administration.

Fourth Article

The fourth article was Netanyahu Apologizes After Blaming Security Chiefs for Failure in Hamas Attack published on October 29th, 2023. The article has 645 words and was published to the world news/middle east section. We separated this article by sentence.

```
fourth_txt= netan_dat_txt$full_txt[4]
class(fourth_txt)
```

```
## [1] "list"
```

```
fourth_txt=fourth_txt[[1]]
tot_fourth_len=length(fourth_txt)

#getting each sentence
fourth_article_sentences=c()


for (i in 1:tot_fourth_len){
  if(fourth_txt[i] != ""){
    fourth_article_sentences=c(fourth_article_sentences,fourth_txt[i])}
}
```

```r
#all Paragraphs of the 4th article
#fourth_article_sentences

fourth_article_sentences = lapply(fourth_article_sentences, function(x) unlist(str_split(x, "\\. ")))
fourth_article_sentences = unlist(fourth_article_sentences)


fourth_df = tibble(sentence = 1:length(fourth_article_sentences), text = fourth_article_sentences)

#obj_4 = cnlp_annotate(fourth_df$text)

tidy_art4 <- fourth_df %>%
  group_by(sentence) %>%
  ungroup() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```r
#NRC Results fourth article
sar_4_nrc = fourth_df %>%
  unnest_tokens(word, text) %>%
  inner_join(nrc_sentiments, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relations
## i Row 7 of `x` matches multiple rows in `y`.
## i Row 13510 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

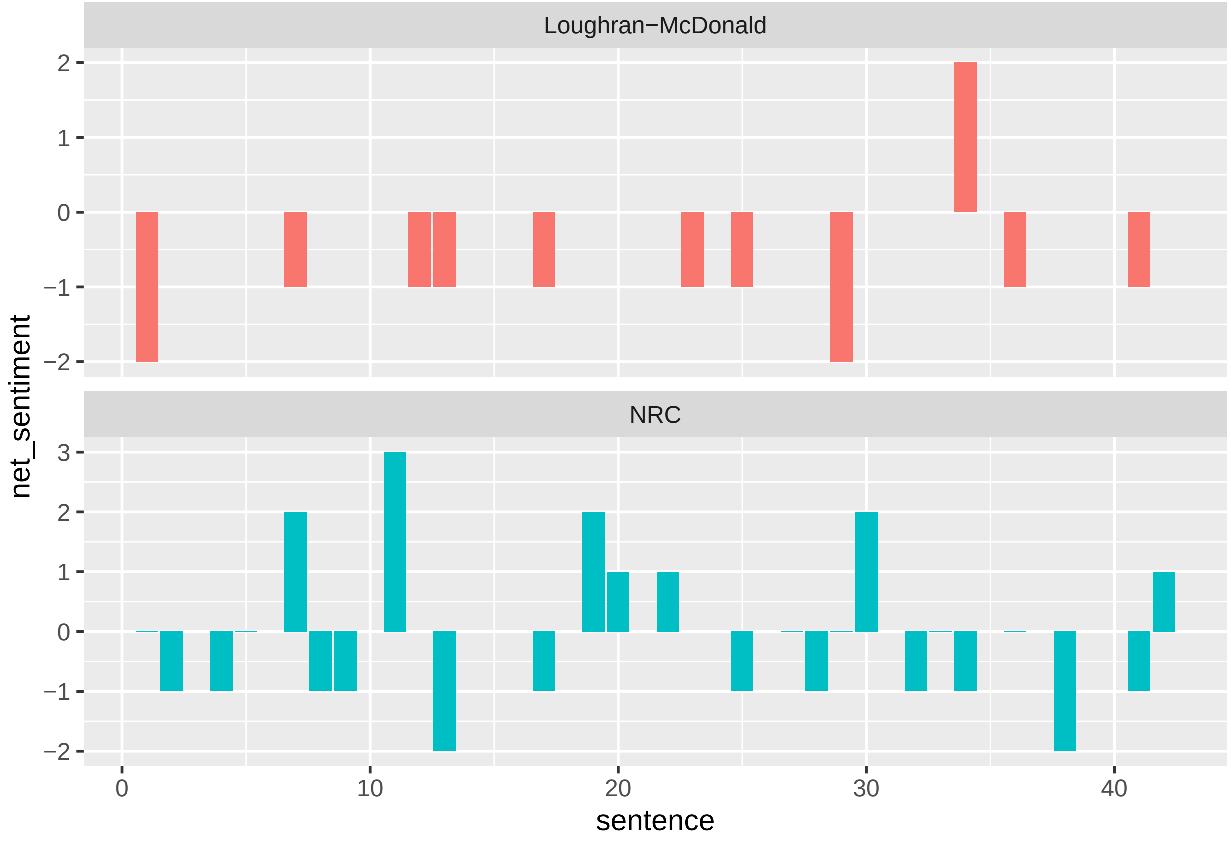
```r
# Loughran results for 4th article
sar_4_loughran <- tidy_art4 %>%
  inner_join(loughran, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

```r
nrc_and_loughran_art4 <- bind_rows(
  tidy_art4 %>%
    inner_join(nrc_pos_and_neg) %>%
    mutate(method = "NRC"),
  tidy_art4 %>%
    inner_join(loughran_pos_and_neg) %>%
    mutate(method = "Loughran-McDonald")) %>%
  group_by(sentence, sentiment, method) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Joining with `by = join_by(word)`
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'sentence', 'sentiment'. You can override
## using the `.groups` argument.
```

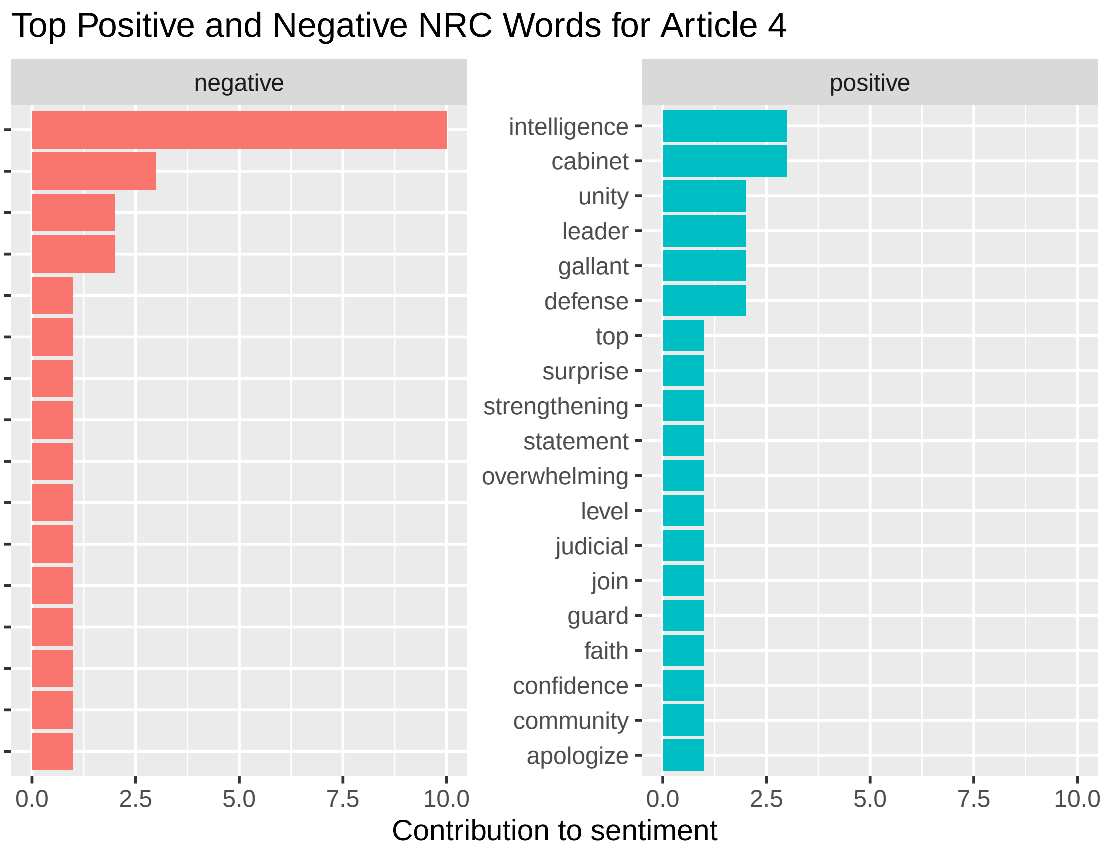
```
# Comparison between the NRC & Loughran sentiment lexicons for article 4
nrc_and_loughran_art4 %>%
  ggplot(aes(sentence, net_sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```



```
art4_nrc_word_counts <- tidy_art4 %>%
  inner_join(nrc_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

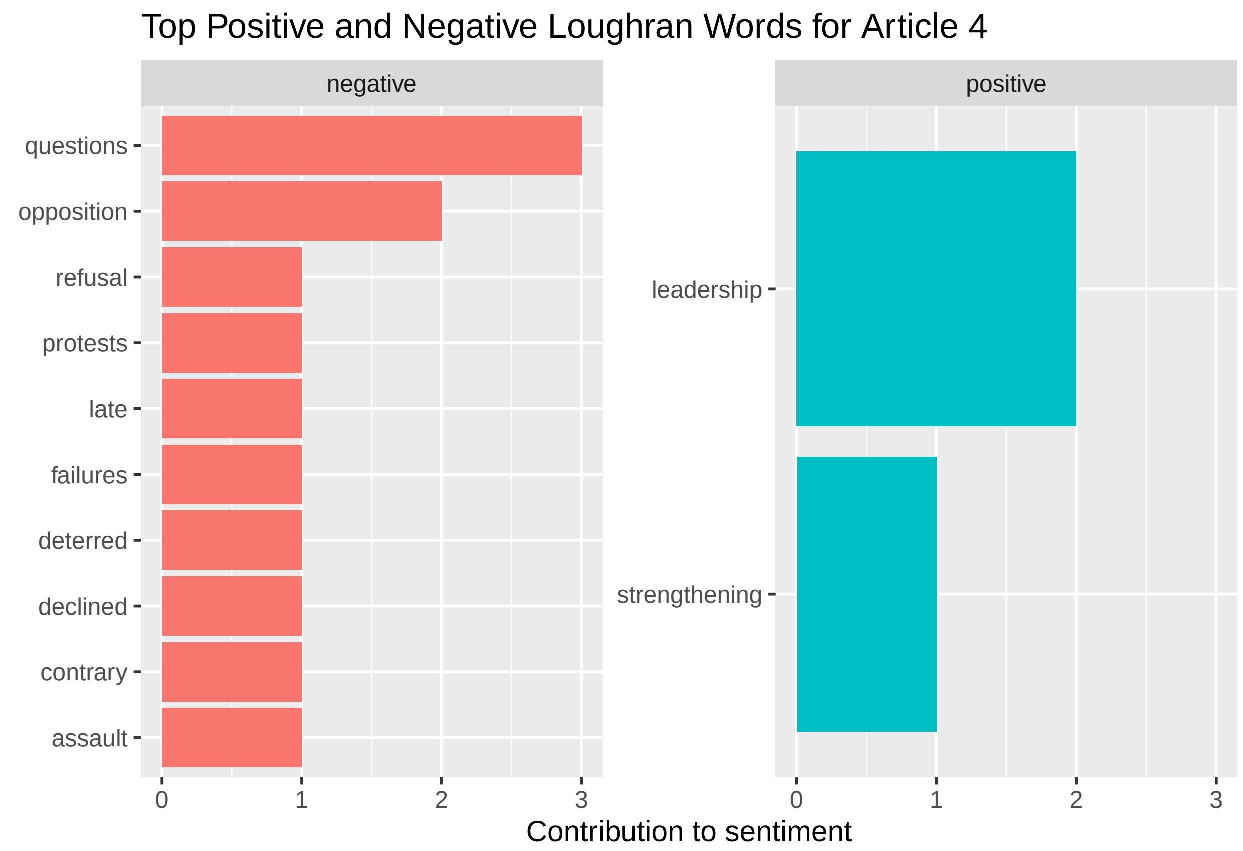
```
art4_nrc_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative NRC Words for Article 4")
```



```
art4_loughran_word_counts <- tidy_art4 %>%
  inner_join(loughran_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
art4_loughran_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) +
  ggtitle("Top Positive and Negative Loughran Words for Article 4")
```



I think the `nrc` lexicon did a good job picking up the sentiment for this article, although the `loughran` is more accurate still. A key point of the article was Mr. Netanyahu's deflection of blame for the attack towards members of his government. While the article spoke about him apologizing for some of his statements, there was still a general negative sentiment with criticism from members of his own government (in the article).

Fifth Article

The fifth article was Netanyahu Finds Himself at War in Gaza and at Home which has 1196 words. The article was published on October 29th, 2023 to the world news/middle east section. We separated this article by sentence.

```
fifth_txt= netan_dat_txt$full_txt[5]
class(fifth_txt)
```

```
## [1] "list"
```

```
fifth_txt=fifth_txt[[1]]
tot_fifth_len=length(fifth_txt)

#getting each sentence
fifth_article_sentences=c()


for (i in 1:tot_fifth_len){
  if(fifth_txt[i] != ""){
    fifth_article_sentences=c(fifth_article_sentences,fifth_txt[i])}
}
```

```
#all Paragraphs of the 5th article
#fifth_article_sentences

fifth_article_sentences = lapply(fifth_article_sentences, function(x) unlist(str_split(x, "\\. ")))
fifth_article_sentences = unlist(fifth_article_sentences)


fifth_df = tibble(sentence = 1:length(fifth_article_sentences), text = fifth_article_sentences)

#obj_5 = cnlp_annotate(fifth_df$text)

tidy_art5 <- fifth_df %>%
  group_by(sentence) %>%
  ungroup() %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```
#NRC Results fifth article
sar_5_nrc = fifth_df %>%
  unnest_tokens(word, text) %>%
  inner_join(nrc_sentiments, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relationsh
## i Row 35 of `x` matches multiple rows in `y`.
## i Row 13510 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

```
# Loughran results for 5th article
sar_5_loughran <- tidy_art5 %>%
  inner_join(loughran, by = "word") %>%
  group_by(sentence, sentiment) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Warning in inner_join(., loughran, by = "word"): Detected an unexpected many-to-many relationship be
## i Row 35 of `x` matches multiple rows in `y`.
## i Row 2296 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## `summarise()` has grouped output by 'sentence'. You can override using the
## `.groups` argument.
```

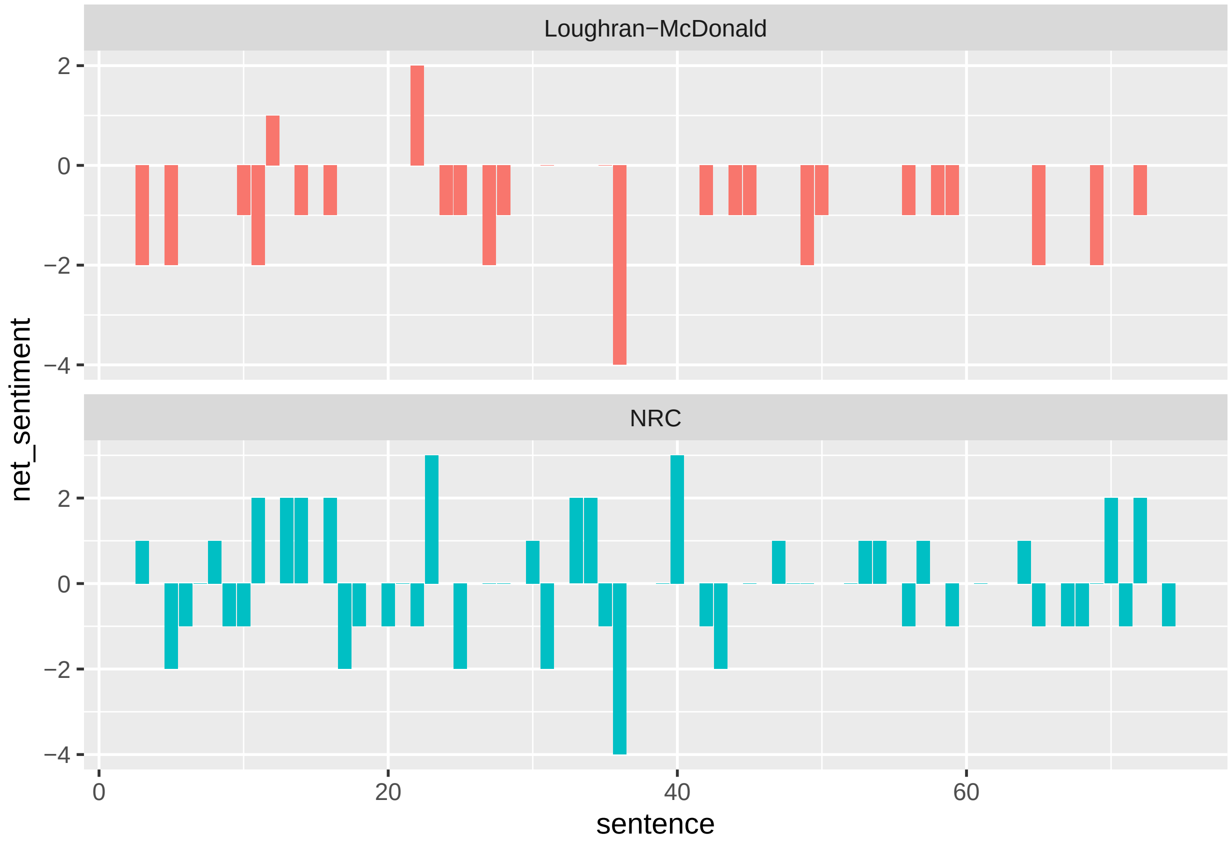
```
nrc_and_loughran_art5 <- bind_rows(
  tidy_art5 %>%
    inner_join(nrc_pos_and_neg) %>%
    mutate(method = "NRC"),
  tidy_art5 %>%
    inner_join(loughran_pos_and_neg) %>%
    mutate(method = "Loughran-McDonald")) %>%
  group_by(sentence, sentiment, method) %>%
  summarise(sentiment_count = n()) %>%
  spread(key = sentiment, value = sentiment_count, fill = 0) %>%
  mutate(net_sentiment = positive - negative)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., nrc_pos_and_neg): Detected an unexpected many-to-many relationship between
## i Row 446 of `x` matches multiple rows in `y`.
## i Row 5459 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Joining with `by = join_by(word)`
## `summarise()` has grouped output by 'sentence', 'sentiment'. You can override
## using the `.groups` argument.
```

```
# Comparison between the NRC & Loughran sentiment lexicons for article 5
nrc_and_loughran_art5 %>%
  ggplot(aes(sentence, net_sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```

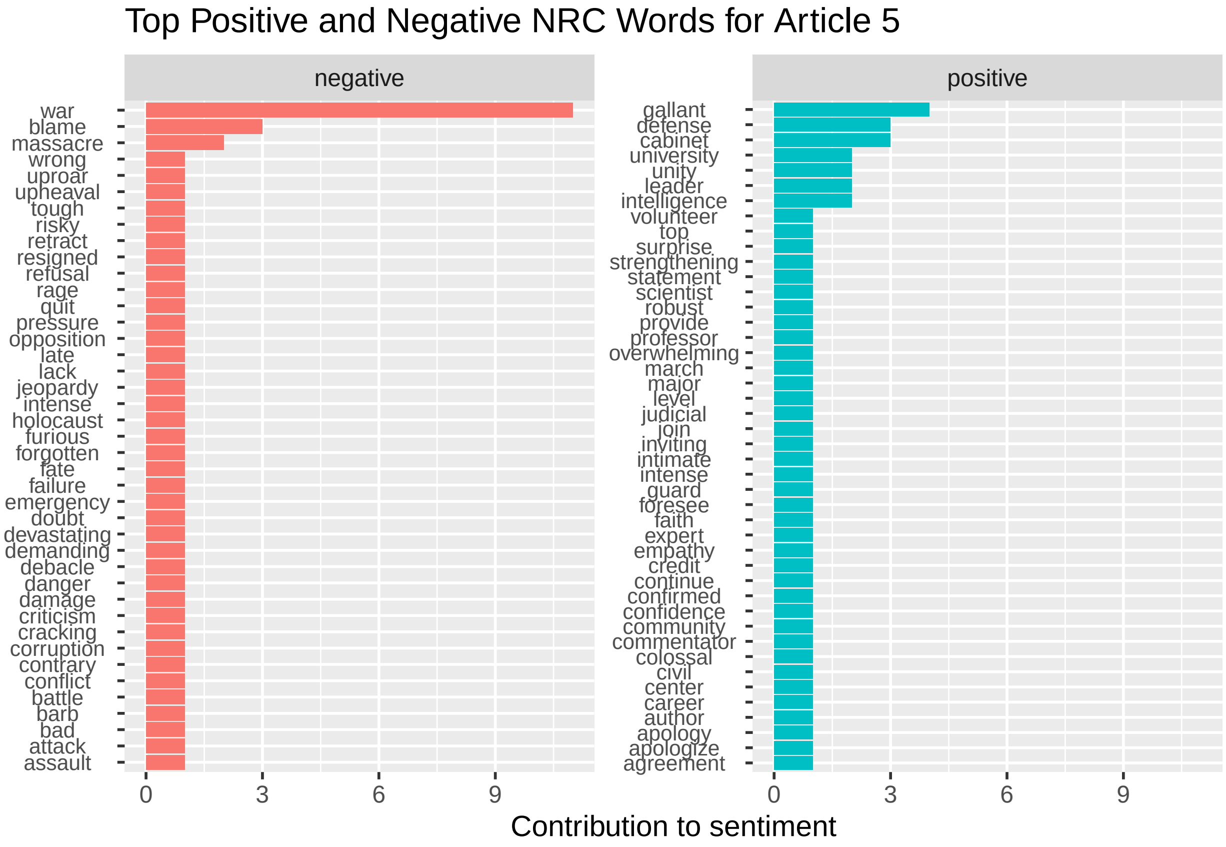


```
art5_nrc_word_counts <- tidy_art5 %>%
  inner_join(nrc_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in inner_join(., nrc_pos_and_neg): Detected an unexpected many-to-many relationship between
## i Row 446 of `x` matches multiple rows in `y`.
## i Row 5459 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

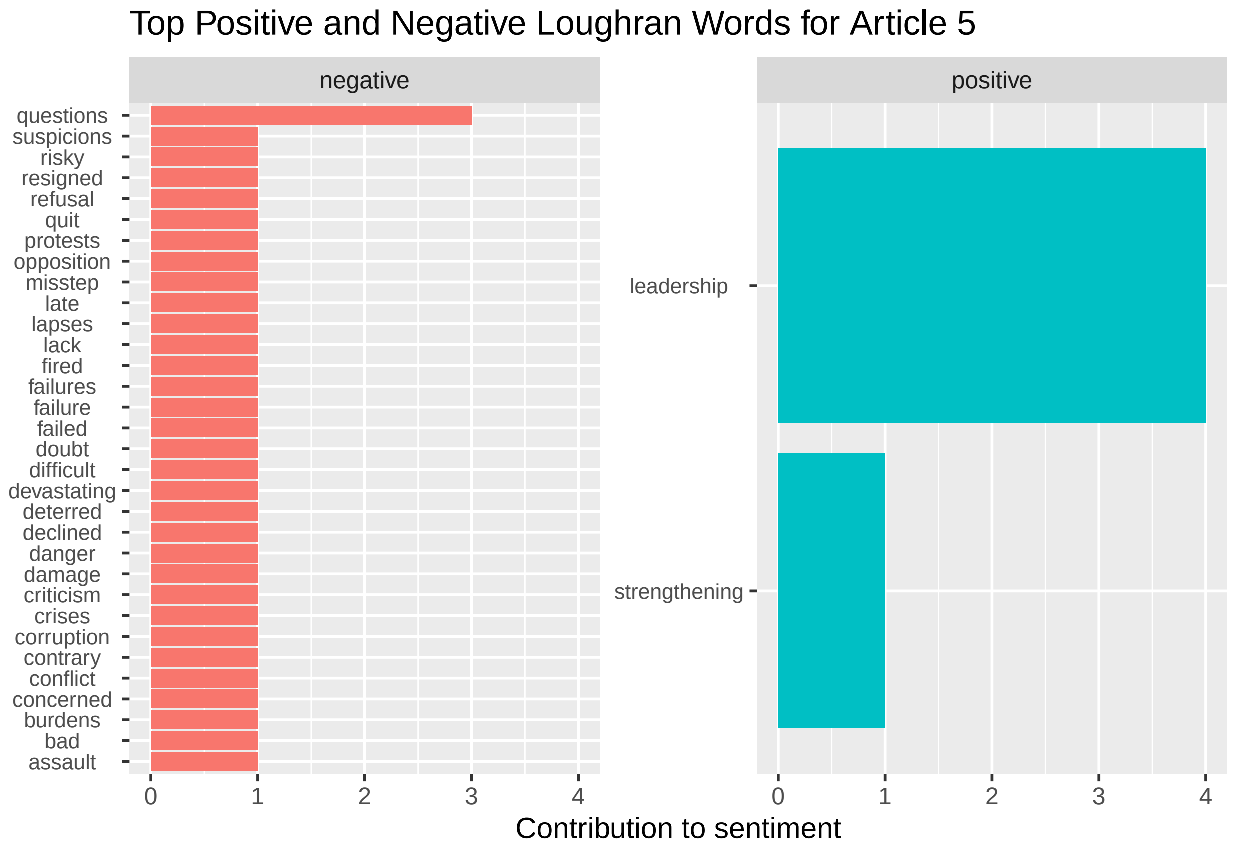
```
art5_nrc_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  theme(axis.text.y = element_text(angle = 0, hjust = .5, size = 8)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative NRC Words for Article 5")
```



```
art5_loughran_word_counts <- tidy_art5 %>%
  inner_join(loughran_pos_and_neg) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
## Joining with `by = join_by(word)`
```

```
art5_loughran_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  theme(axis.text.y = element_text(angle = 0, hjust = .5, size = 8)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment", y = NULL) +
  ggtitle("Top Positive and Negative Loughran Words for Article 5")
```

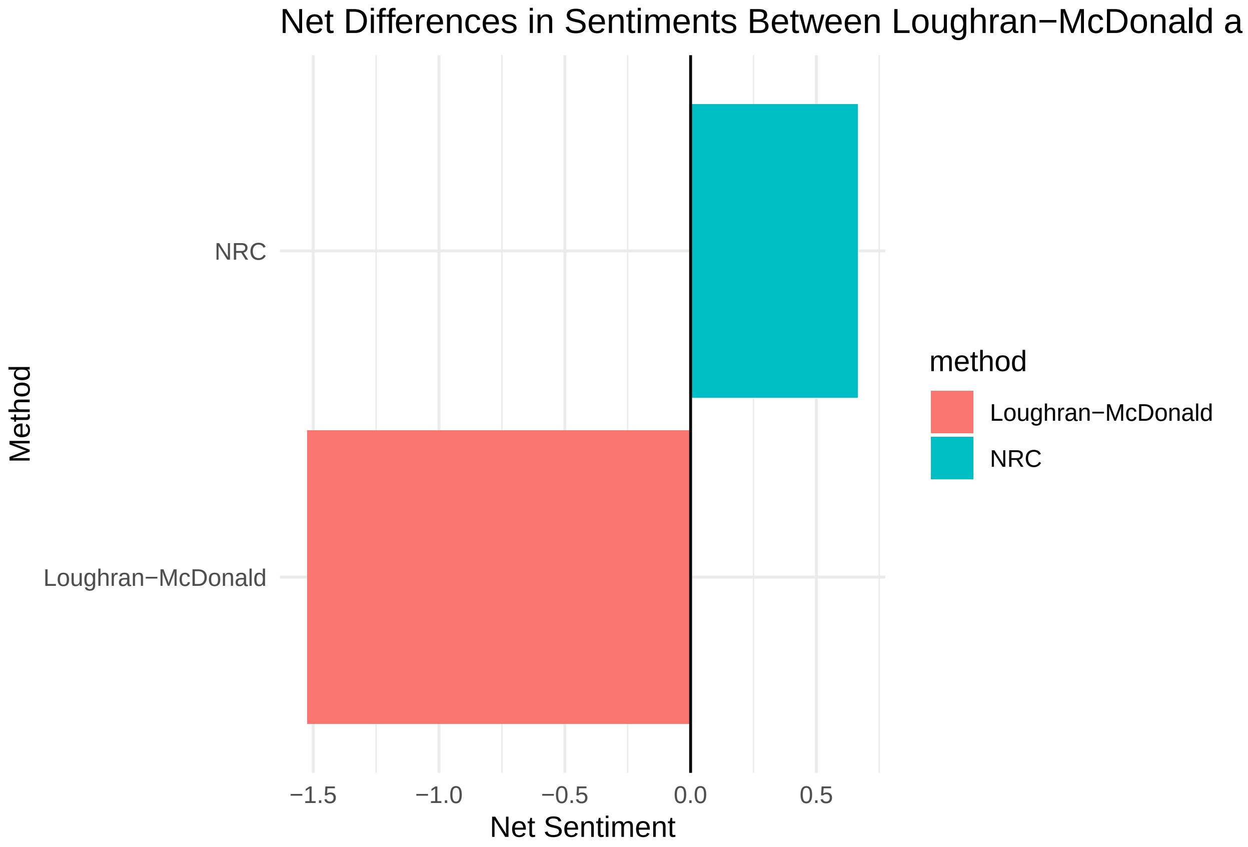


The `loughran` did a good job at getting the sentiment for this article.

Additional Analyses

Finally, let's wrap up by analyzing net sentiment means. First, we will start off by looking at the mean differences between NRC and Loughran sentiments.

```
nrc_and_loughran_art1_id <- nrc_and_loughran_art1 %>% subset(select = -paragraph) %>%
  mutate(ID = "Article 1")
nrc_and_loughran_art2_id <- nrc_and_loughran_art2 %>% subset(select = -sentence) %>%
  mutate(ID = "Article 2")
nrc_and_loughran_art3_id <- nrc_and_loughran_art3 %>% subset(select = -sentence) %>%
  mutate(ID = "Article 3")
nrc_and_loughran_art4_id <- nrc_and_loughran_art4 %>% subset(select = -sentence) %>%
  mutate(ID = "Article 4")
nrc_and_loughran_art5_id <- nrc_and_loughran_art5 %>% subset(select = -sentence) %>%
  mutate(ID = "Article 5")

nrc_and_loughran_full <- rbind(nrc_and_loughran_art1_id, nrc_and_loughran_art2_id,
                                nrc_and_loughran_art3_id, nrc_and_loughran_art4_id,
                                nrc_and_loughran_art5_id)


nrc_and_loughran_means <- nrc_and_loughran_full %>% group_by(method) %>% summarize(means = mean(net_sen

nrc_and_loughran_means %>%
  ggplot(aes(method, means, fill = method)) +
  geom_col(position = "stack") +
  geom_hline(yintercept = 0, linetype = "solid") +
  labs(title = "Net Differences in Sentiments Between Loughran-McDonald and NRC Analyses", x = "Method"
  theme_minimal()
```



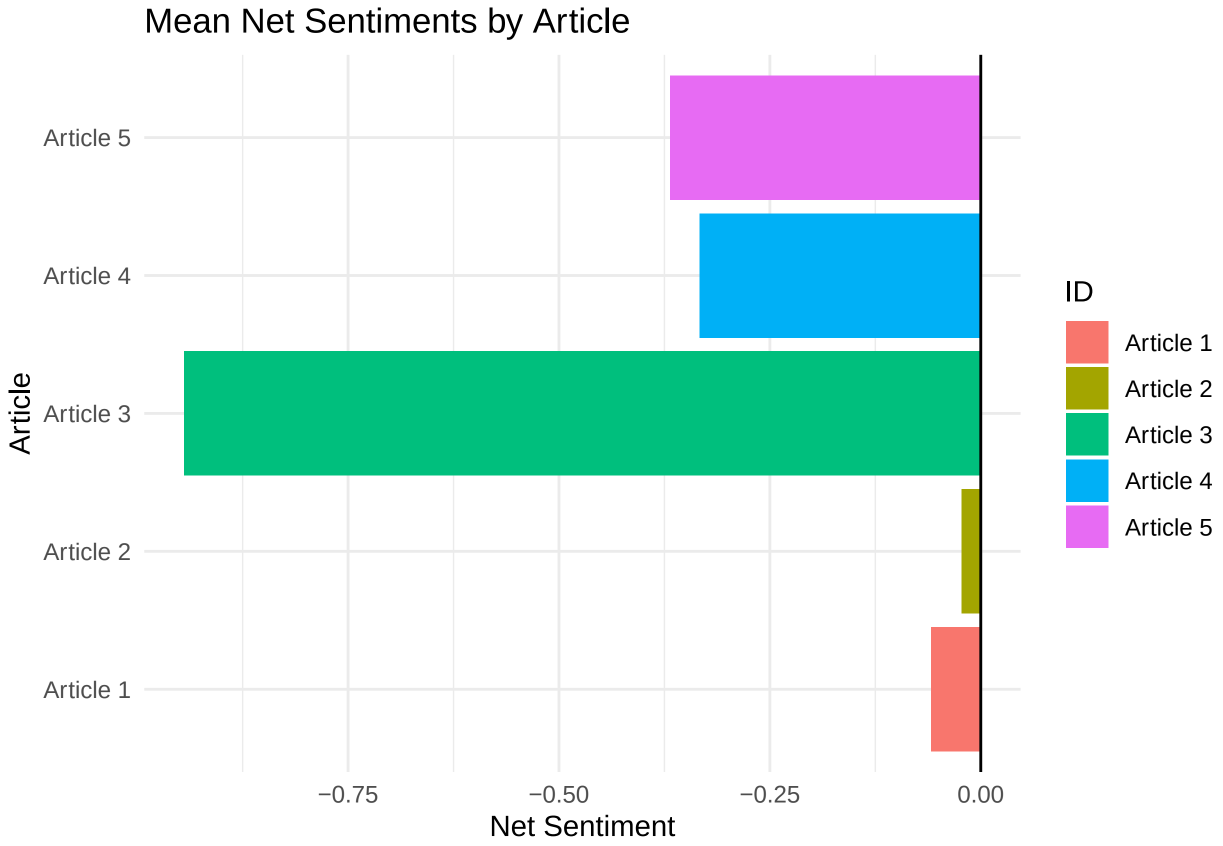
As we can see, it appears that, overall, the NRC analysis coded the article sentiments more neutrally, on average, than the Loughran analysis.

And now, let's look at the sentiments across all five articles

```
nrc_and_loughran_means <- nrc_and_loughran_full %>% group_by(ID) %>% summarize(means = mean(net_sentimen

kable(nrc_and_loughran_means)
```

ID	means
Article 1	-0.0588235
Article 2	-0.0227273
Article 3	-0.9444444
Article 4	-0.3333333
Article 5	-0.3684211

```
nrc_and_loughran_means %>%
  ggplot(aes(ID, means, fill = ID)) +
  geom_col(position = "stack") +
  geom_hline(yintercept = 0, linetype = "solid") +
  labs(title = "Mean Net Sentiments by Article", x = "Article", y = "Net Sentiment") + coord_flip() + t
```



As we can see here, it appears that each article was coded, on average, with a greater degree of negative sentiments than positive sentiments. Furthermore, it would appear that the 3rd article was coded as the most negative article of the five, whereas the 2nd article was coded as the least negative

Now, let's look at the sentiments across the five articles by the analytic method used:

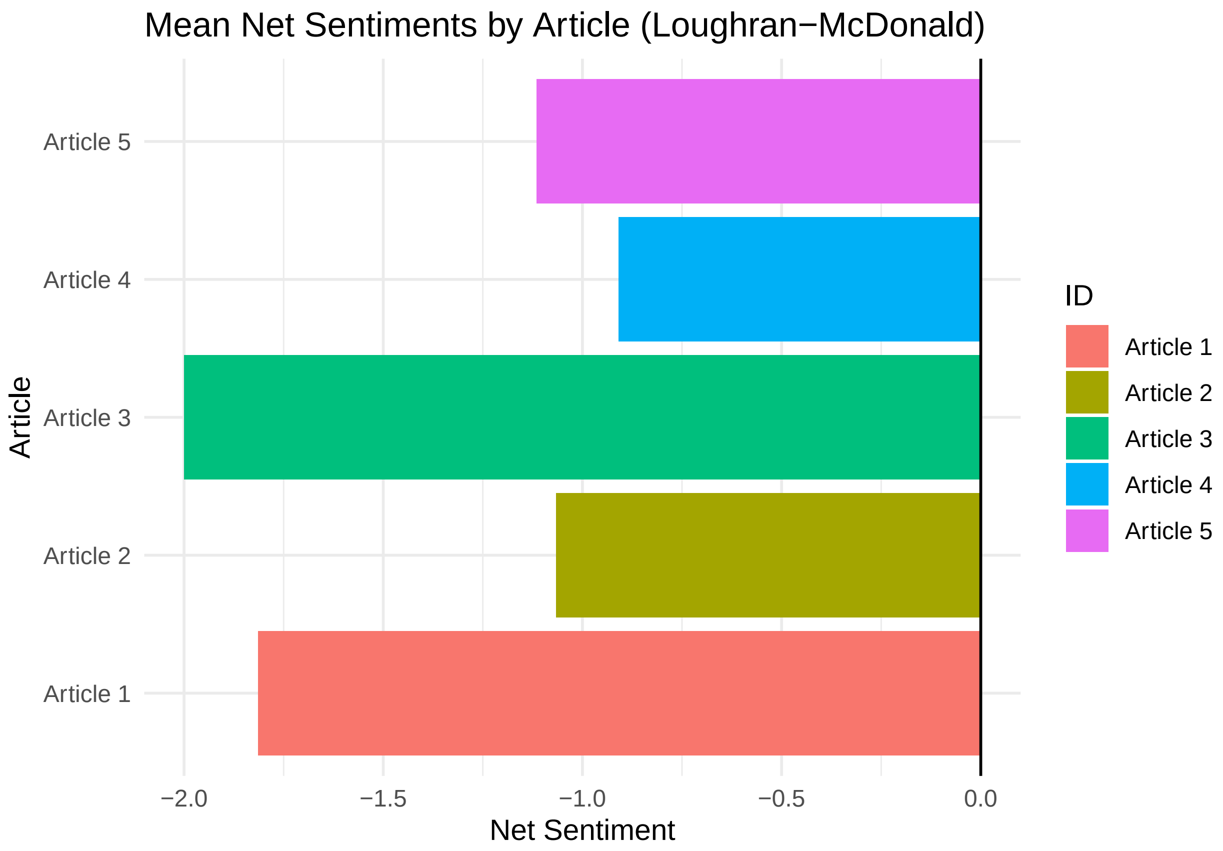
```
loughran_means <- nrc_and_loughran_full %>% filter(method == "Loughran-McDonald") %>% group_by(ID) %>%

kable(loughran_means)
```

ID	means
Article 1	-1.8142857
Article 2	-1.0666667
Article 3	-2.0000000
Article 4	-0.9090909
Article 5	-1.1153846

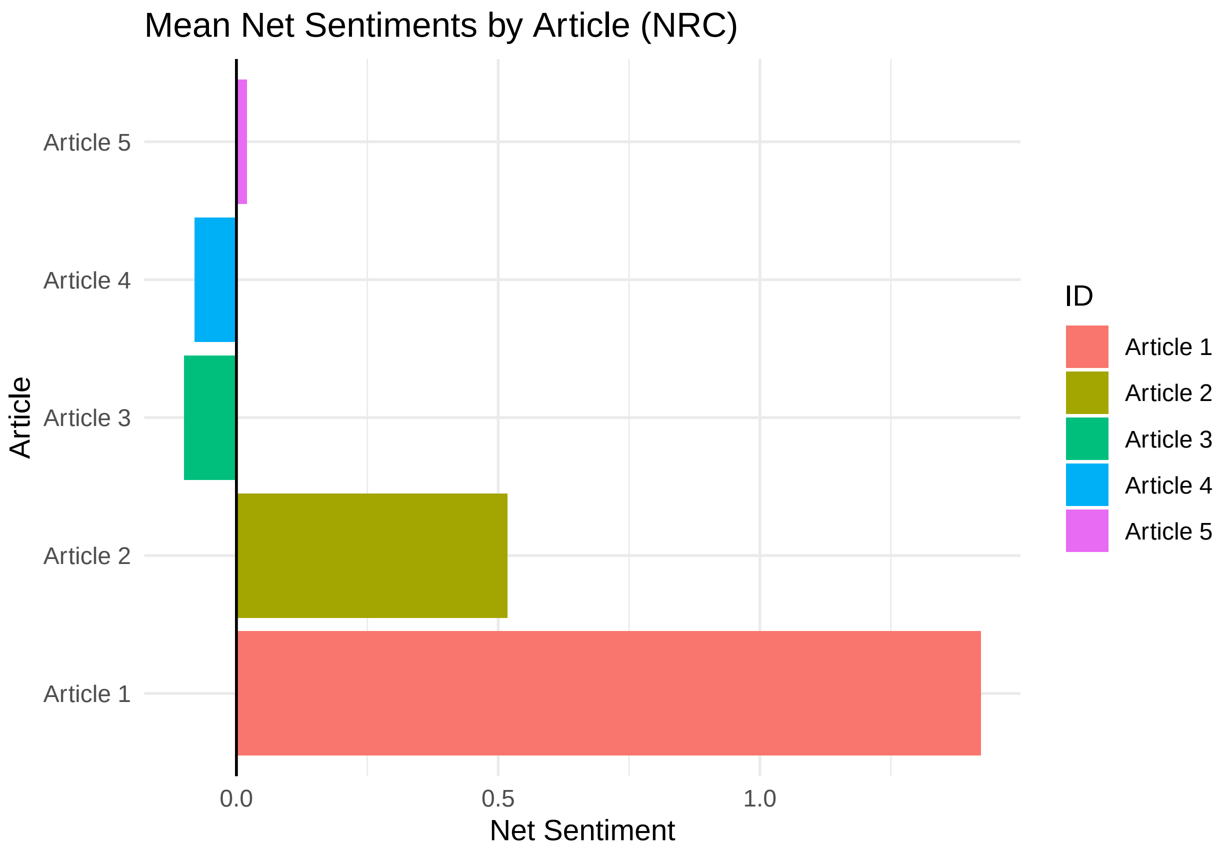
```
nrc_means <- nrc_and_loughran_full %>% filter(method == "NRC") %>% group_by(ID) %>% summarize(means = me

kable(nrc_means)
```

ID	means
Article 1	1.4216867
Article 2	0.5172414
Article 3	-0.1000000
Article 4	-0.0800000
Article 5	0.0200000

```
loughran_means %>%
  ggplot(aes(ID, means, fill = ID)) +
  geom_col(position = "stack") +
  geom_hline(yintercept = 0, linetype = "solid") +
  labs(title = "Mean Net Sentiments by Article (Loughran-McDonald)", x = "Article", y = "Net Sentiment"
  theme_minimal()
```



```
nrc_means %>%
  ggplot(aes(ID, means, fill = ID)) +
  geom_col(position = "stack") +
  geom_hline(yintercept = 0, linetype = "solid") +
  labs(title = "Mean Net Sentiments by Article (NRC)", x = "Article", y = "Net Sentiment") + coord_flip
```



It would appear that the Loughran-McDonald analysis, as expected, coded the articles in a much more negative manner than the NRC analysis. In addition, the Loughran-McDonald analysis identified Article 1 as containing the greatest amount of negative sentiments. The NRC analysis, on the otherhand, identified far more positive sentiments, overall. Surprisingly, in contrast with the Loughran-McDonald analysis, the NRC analysis coded Article 1 with the highest average positive sentiments.

Conclusion and Future Directions

In conclusion, sentiment analysis can be used to analyze the sentiment of text. In the context of this assignment, we used sentiment analysis to analyze what type of sentiment New York Times article written about Israeli Prime Minister Benjamin Netanyahu has.

Interestingly, both analyses tended to differ with regards to their sentiment coding schemas per article. These differences are most apparent for article 1, wherein the NRC method captured a more net positive outlook than the Loughran-McDonald method. These findings demonstrate the disparity in sentiment coding methods across lexicons, and highlight the importance of human or AI intervention to determine the most appropriate model for their purposes.

One way to enhance this project moving forward is to compare the results of both `nrc` and `loughran` for all articles by performing statistical analyses to see which lexicon is better at getting the sentiment for these types of articles. Another interesting enhancement to this project is to track down the sentiment towards

Mr.Netanyahu over time and see how it changes based on world events. Sentiment analysis is a valuable tool to gauge the opinion of groups.

Disclaimer: The sentiment analysis conducted in this assignment is a **PURLEY** academic exercise aimed at understanding and applying data science techniques. It is **NOT** intended to promote, endorse, or engage in any form of criticism against any individual or government, including the Israeli government. This analysis should **NOT** be misconstrued or misinterpreted as a political statement or an act of anti-semitism. As data science students and American citizens, we exercise our protected right to conduct and discuss factual, evidence-based research on public figures and world leaders in accordance with ethical and moral standards. Our commitment as data scientists is to report accurately, use truthful data, and avoid misleading representations in all our work.

Resources Used

tidytextmining

NYT API Developer Documentation