Text Mining and Sentiment Analysis

John Cruz

2023-04-01

Introduction

Text Mining with R, Chapter 2, looks at sentiment analysis. The authors provide an example using the text of Jane Austen's six completed, published novels from the *janeaustenr* library. All the code is originally credited to the authors, unless otherwise noted.

Required Libraries

```
library(tidyverse)
library(tidytext)
library(janeaustenr)
library(stringr)
library(jsonlite)
library(glue)
library(lubridate)
library(ggrepel)
```

Tidy Up Jane Austen's Work

The authors take the text of the novels and converts the text to the tidy format using *unnest_tokens()*. They also create other columns to keep track of which line and chapter of the book each word comes from.

```
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)

knitr::kable(head(tidy_books), caption = "Brief View of Tokenized Words")
```

Table 1: Brief View of Tokenized Words

book	linenumber	chapter	word
Sense & Sensibility	1	0	sense
Sense & Sensibility	1	0	and
Sense & Sensibility	1	0	sensibility
Sense & Sensibility	3	0	by
Sense & Sensibility	3	0	jane
Sense & Sensibility	3	0	austen

Determining Overall Sentiment

Next, count up how many positive and negative words there are in defined sections of each book, along with a net sentiment score. They define an index here to keep track of where they are in the narrative. The index counts up sections of 80 lines of text.

```
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)

knitr::kable(head(jane_austen_sentiment), caption = "Brief View of Sentiment Scores by Indexing")
```

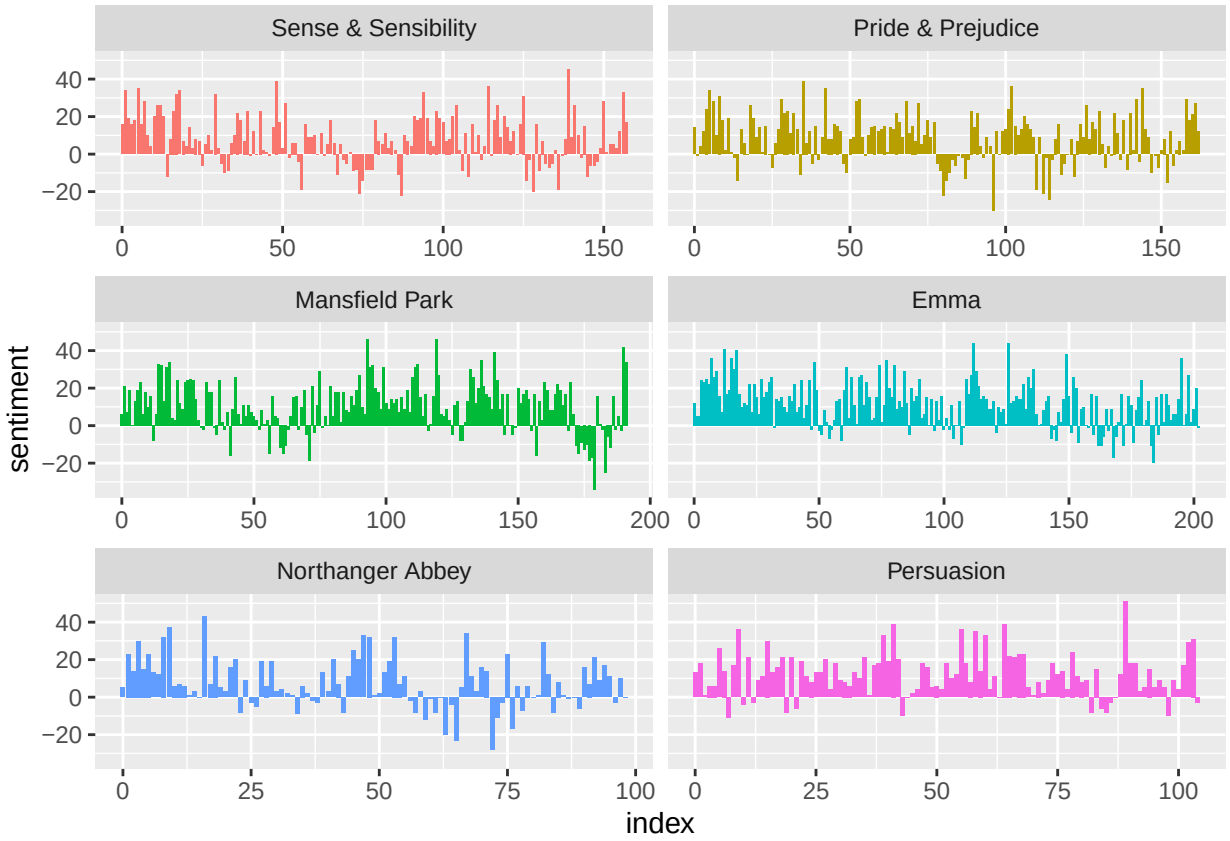
Table 2: Brief View of Sentiment Scores by Indexing

book	index	negative	positive	sentiment
Sense & Sensibility	0	16	32	16
Sense & Sensibility	1	19	53	34
Sense & Sensibility	2	12	31	19
Sense & Sensibility	3	15	31	16
Sense & Sensibility	4	16	34	18
Sense & Sensibility	5	16	51	35

Visualizing Sentiment throughout each Novel

Finally, plot how each novel changes toward more positive or negative sentiment over the trajectory of the story.

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



Which Chapter has the Most Negative Words?

The authors also provide a proportion table to determine which chapter has the most negative words in each book.

```
bingnegative <-
  get_sentiments("bing") |>
  filter(sentiment == "negative")

wordcounts <- tidy_books |>
  group_by(book, chapter) |>
  summarize(words = n())

ratio_tbl <-
  tidy_books |>
  semi_join(bingnegative) |>
  group_by(book, chapter) |>
  summarize(negativewords = n()) |>
  left_join(wordcounts, by = c("book", "chapter")) |>
  mutate(ratio = negativewords/words) |>
  filter(chapter != 0) |>
  slice_max(ratio, n = 1) |>
  ungroup()

knitr::kable(ratio_tbl)
```

book	chapter	negativewords	words	ratio
Sense & Sensibility	43	161	3405	0.0472834
Pride & Prejudice	34	111	2104	0.0527567
Mansfield Park	46	173	3685	0.0469471
Emma	15	151	3340	0.0452096
Northanger Abbey	21	149	2982	0.0499665
Persuasion	4	62	1807	0.0343110

Note: All work from this point forward has been created by me.

Corpus: NY Times Articles

Lets look at the NY Times published articles in March 2023. However, I will primarily look at the lead paragraph of each article. The goal is to get an idea of what kind of sentiment is being used within the different sections of each paragraph they offer such as Arts, U.S. and Sports. I will also look into seeing if certain times of the day lends itself more to positive or negative sentiments.

Connect to NY Times API

```
api_cnxn <-
  fromJSON(glue("https://api.nytimes.com/svc/archive/v1/2023/3.json?api-key={rstudioapi::askForPassword

ny_times <-
    as.data.frame(api_cnxn) |>
    janitor::clean_names()

write_csv(ny_times, 'ny_times.csv')
```

Tidy Up Article Data

Clean up column formatting for published dates.

```
section_df <-
  ny_times |>
  select(response_docs_pub_date, response_docs_section_name, response_docs_lead_paragraph) |>
  mutate(response_docs_pub_date = str_extract(response_docs_pub_date, "[:graph:]*(?=\\+)")) |>
  rename(pub_date = response_docs_pub_date, lead_paragraph = response_docs_lead_paragraph, section = res

section_df$pub_date <-
  section_df$pub_date |>
  ymd_hms()

section_df$hour <-
  section_df$pub_date |>
  hour()
```

Tokenize Sections

```
tokenize_df <-
  section_df |>
  unnest_tokens(word, lead_paragraph)

knitr::kable(head(tokenize_df))
```

pub_date	section	hour	word
2023-03-01 00:00:07	Opinion	0	to
2023-03-01 00:00:07	Opinion	0	president
2023-03-01 00:00:07	Opinion	0	emmanuel
2023-03-01 00:00:07	Opinion	0	macron
2023-03-01 00:00:07	Opinion	0	of
2023-03-01 00:00:07	Opinion	0	france

Categorize Sentiments between Sections

Here, we can see that the most negative overall sentiment is the U.S. section at 71%. The most positive overall sentiment is Arts at 22%.

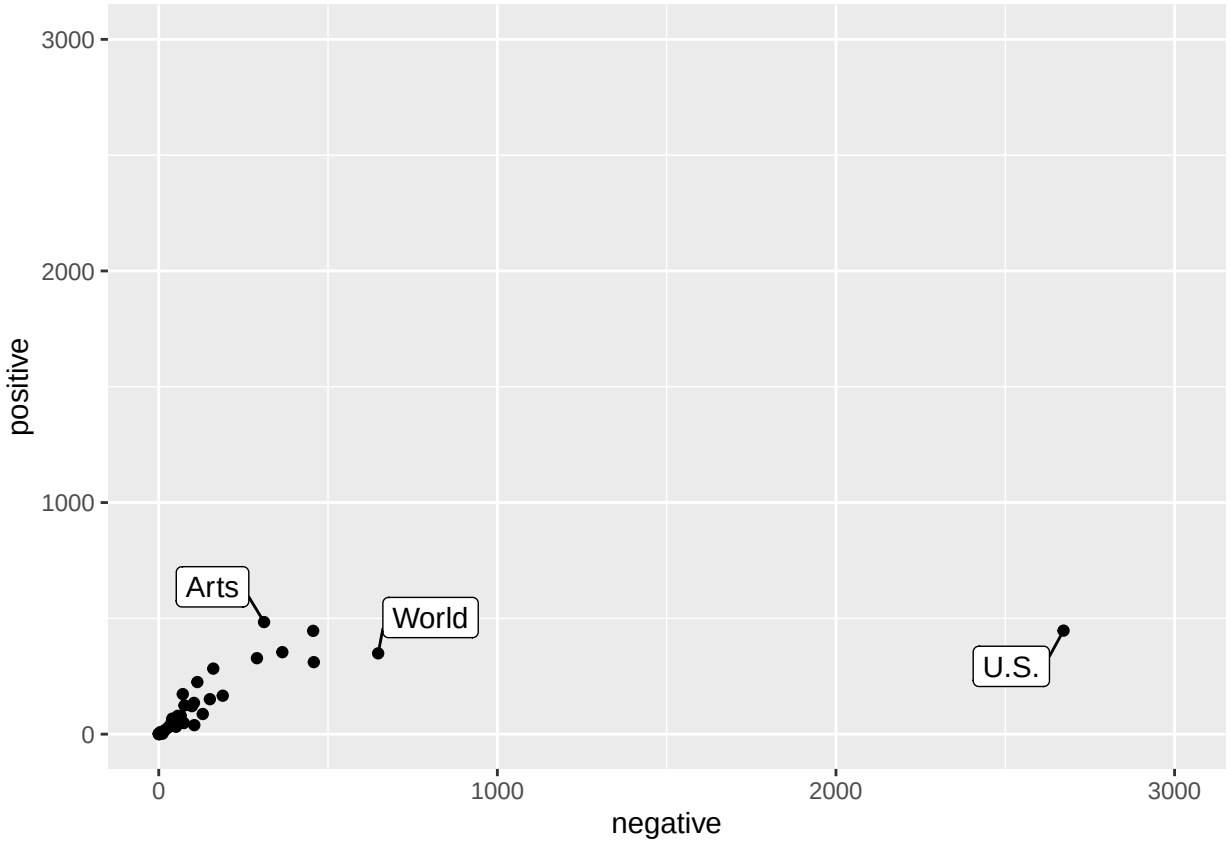
```
sentiment_df <-
  tokenize_df |>
  inner_join(get_sentiments("bing")) |>
  count(section, sentiment) |>
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) |>
  mutate(total_words = negative + positive,
         ovr_sentiment = positive - negative,
        pct = round(ovr_sentiment/total_words *100, 2))

knitr::kable(sentiment_df, caption = "Overall Sentiment based on Section")
```

Table 5: Overall Sentiment based on Section

section	negative	positive	total_words	ovr_sentiment	pct
Admin	1	0	1	-1	-100.00
Arts	311	484	795	173	21.76
Books	365	354	719	-11	-1.53
Briefing	130	87	217	-43	-19.82
Business Day	458	311	769	-147	-19.12
Climate	28	29	57	1	1.75
Corrections	51	32	83	-19	-22.89
Crosswords & Games	104	135	239	31	12.97
Education	11	2	13	-9	-69.23
Fashion & Style	5	5	10	0	0.00
Food	71	173	244	102	41.80
Health	105	39	144	-66	-45.83
Magazine	151	151	302	0	0.00
Movies	290	328	618	38	6.15

section	negative	positive	total_words	ovr_sentiment	pct
New York	189	166	355	-23	-6.48
Obituaries	7	8	15	1	6.67
Opinion	456	446	902	-10	-1.11
Podcasts	39	56	95	17	17.89
Real Estate	56	79	135	23	17.04
Science	74	48	122	-26	-21.31
Smarter Living	0	2	2	2	100.00
Sports	161	283	444	122	27.48
Style	114	225	339	111	32.74
T Brand	0	1	1	1	100.00
T Magazine	76	124	200	48	24.00
Technology	40	67	107	27	25.23
The Learning Network	65	79	144	14	9.72
The Upshot	5	9	14	4	28.57
Theater	98	121	219	23	10.50
Times Insider	18	17	35	-1	-2.86
Travel	38	49	87	11	12.64
U.S.	2672	447	3119	-2225	-71.34
Video	1	0	1	-1	-100.00
Well	52	39	91	-13	-14.29
World	648	349	997	-299	-29.99
Your Money	4	4	8	0	0.00

```
sentiment_df |>
  ggplot(aes(x = negative, y = positive, label = section)) +
  geom_point() +
  geom_label_repel(box.padding = 0.35) +
  xlim(0, 3000) +
  ylim(0, 3000)
```



Categorize Sentiments between Time of Day

When categorizing based on time of day, 0600 hour has the most positive leading paragraph sentiment at 31%, while at 1200 hour, it has the largest difference in negative sentiment at 88%.

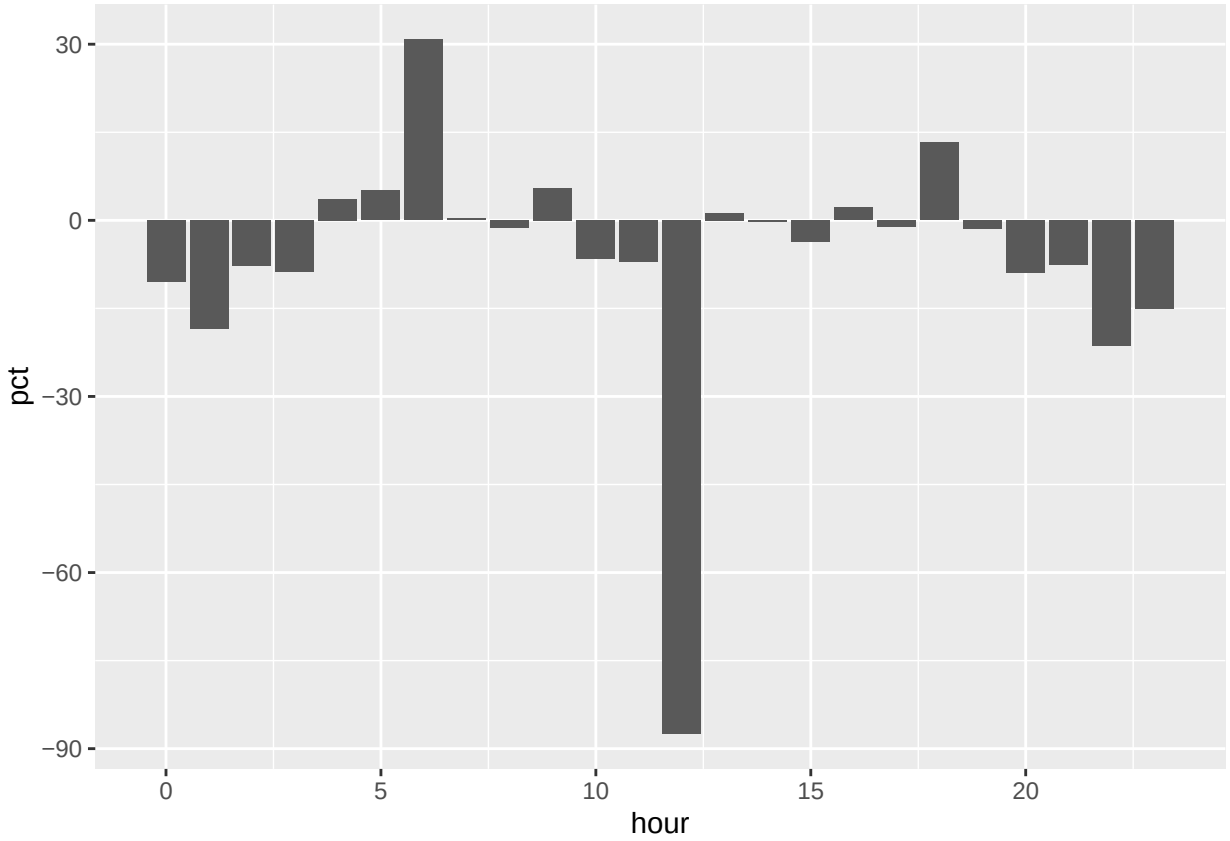
```
pub_date_df <-
  tokenize_df |>
  inner_join(get_sentiments("bing")) |>
  group_by() |>
  count(hour, sentiment) |>
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) |>
  mutate(total_words = negative + positive,
         ovr_sentiment = positive - negative,
        pct = round(ovr_sentiment/total_words *100, 2))

knitr::kable(pub_date_df, caption = "Overall Sentiment based on Hour")
```

Table 6: Overall Sentiment based on Hour

hour	negative	positive	total_words	ovr_sentiment	pct
0	153	124	277	-29	-10.47
1	77	53	130	-24	-18.46
2	77	66	143	-11	-7.69
3	68	57	125	-11	-8.80

hour	negative	positive	total_words	ovr_sentiment	pct
4	107	115	222	8	3.60
5	84	93	177	9	5.08
6	46	87	133	41	30.83
7	163	164	327	1	0.31
8	80	78	158	-2	-1.27
9	749	836	1585	87	5.49
10	554	485	1039	-69	-6.64
11	196	170	366	-26	-7.10
12	2106	140	2246	-1966	-87.53
13	190	195	385	5	1.30
14	209	208	417	-1	-0.24
15	326	303	629	-23	-3.66
16	247	258	505	11	2.18
17	230	225	455	-5	-1.10
18	173	226	399	53	13.28
19	239	232	471	-7	-1.49
20	212	177	389	-35	-9.00
21	221	190	411	-31	-7.54
22	204	132	336	-72	-21.43
23	183	135	318	-48	-15.09

```
pub_date_df |>
  ggplot(aes(x = hour, y = pct)) +
  geom_bar(stat = 'identity')
```



Lexicon Library

Researching for other types of lexicon libraries R has to offer, I came across the **lexicon** package. Here it contains many different methods, one of them which is the *hash_sentiment_senticnet()* function. It is a data.table dataset containing an augmented version of Cambria, Poria, Bajpai,& Schuller's (2016) positive/negative word list as sentiment lookup values. Further documentation can be found here.

```
library(lexicon)

knitr::kable(head(hash_sentiment_senticnet), caption = 'Glance of Lookup Values')
```

Table 7: Glance of Lookup Values

x	y
aaa	0.606
aah	-0.510
abandon	-0.560
abandonment	-0.650
abase	-0.580
abasement	-0.580

Calculate Sentiment based on New Sentiment Lexicon

What we see is that the slope relationship between positive and negative words are much more favorable for positive words when comparing total word counts to each section. This is because I primitively categorized each word based on either being positive or negative without it's strength value accounted for. However, the U.S. section still shows as the leader for all sections with negative words overall.

```
lexicon_df <-
  hash_sentiment_senticnet |>
  rename(word = x, value = y)
```

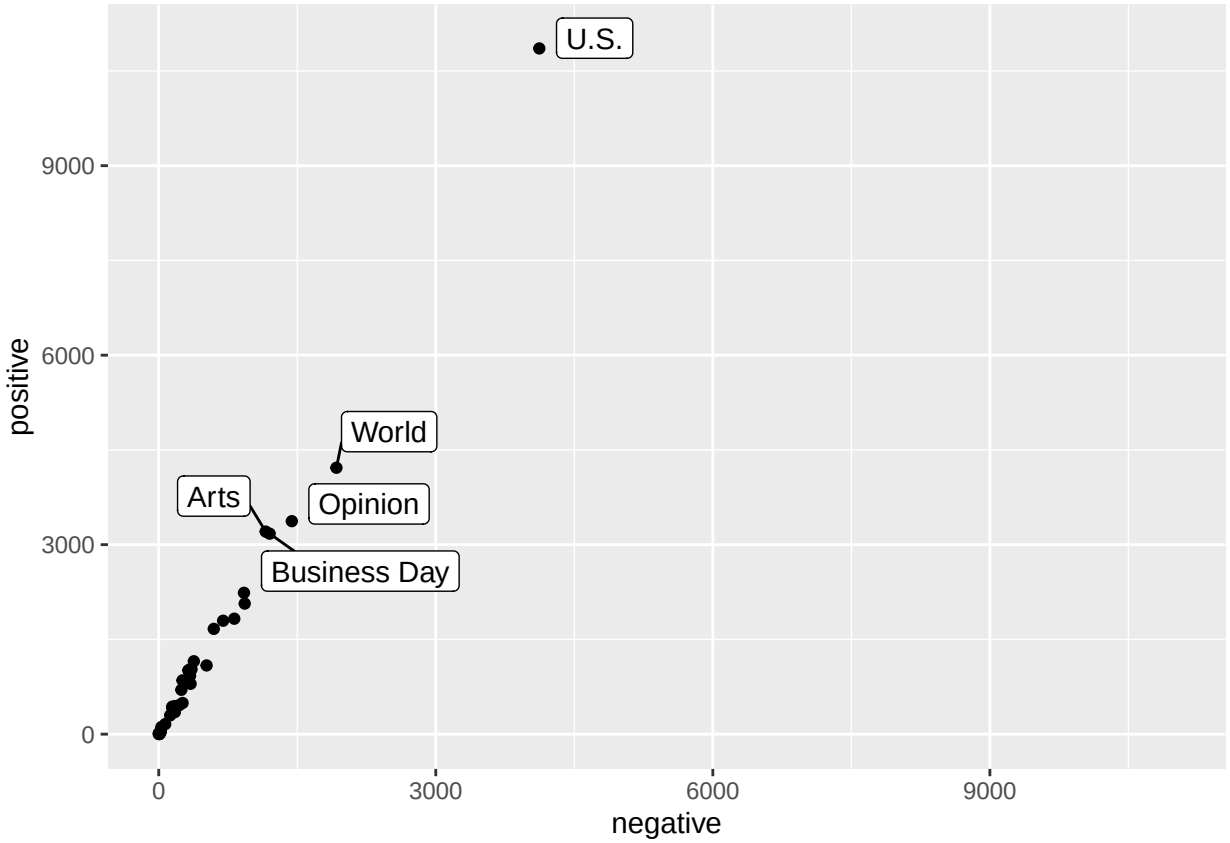
```
lex_sentiment_df <-
  tokenize_df |>
  inner_join(lexicon_df)

lex_section_df <-
  lex_sentiment_df |>
  mutate(type = case_when(value < 0 ~ "negative",
            value > 0 ~ "positive")) |>
  group_by(section, type) |>
  summarise(total_words = n()) |>
  pivot_wider(names_from = type, values_from = total_words) |>
  mutate(total_words = negative + positive)

knitr::kable(lex_section_df, caption = "Overall Sentiment based on Section")
```

Table 8: Overall Sentiment based on Section

section	negative	positive	total_words
Admin	2	1	3
Arts	1158	3207	4365
Books	923	2238	3161
Briefing	339	925	1264
Business Day	1201	3174	4375
Climate	124	298	422
Corrections	146	435	581
Crosswords & Games	344	1031	1375
Education	15	35	50
Fashion & Style	23	38	61
Food	380	1154	1534
Headway	7	2	9
Health	258	494	752
Magazine	518	1088	1606
Movies	931	2067	2998
New York	697	1796	2493
Obituaries	16	35	51
Opinion	1441	3372	4813
Podcasts	147	414	561
Reader Center	5	12	17
Real Estate	320	1010	1330
Science	223	460	683
Smarter Living	2	16	18

section	negative	positive	total_words
Special Series	11	8	19
Sports	818	1828	2646
Style	596	1667	2263
T Brand	NA	4	NA
T Magazine	355	1027	1382
Technology	245	702	947
The Learning Network	257	852	1109
The New York Times Presents	1	4	5
The Upshot	29	113	142
Theater	345	797	1142
Times Insider	70	159	229
Travel	175	444	619
U.S.	4121	10856	14977
Video	6	5	11
Well	175	350	525
World	1924	4218	6142
Your Money	13	30	43

```
lex_section_df |>
  ggplot(aes(x = negative, y = positive, label = section)) +
  geom_point() +
  geom_label_repel(box.padding = 0.35) +
  xlim(0, 11000) +
  ylim(0, 11000)
```



Conclusion

Based on the lexicon sentiment package used, the simple values used can change drastically, however, overall the general tendacies are the same such that the U.S section within in the NY Times generally tends to have more ngative sentiment words being used.