Data 607 Assignment 10A: Sentiment Analysis

Joao De Oliveira

October 29, 2025

Overview

This report reproduces the primary example from Chapter 2: Sentiment Analysis from *Text Mining with R* by **Julia Silge & David Robinson**, then extends it by analyzing a Portuguese book that is one of my personal favorites—**Amor de Perdição, by Camilo Castelo Branco**—and incorporating an additional sentiment lexicon from another R package (Portuguese **OpLexicon** via `lexiconPT`).

Citations

Primary reference (book):
Silge, J., & Robinson, D. (2017). Text Mining with R: A Tidy Approach. O'Reilly. https://www.tidytextmining.com/

Key packages / resources used: `tidyverse`, `tidytext`, `ggplot2`, `janeaustenr`, `gutenbergr` (Gutenberg ID **16425**), `stopwords` (Portuguese stop words), `lexiconPT` (OpLexicon v3.0), `lexicon` (Bing/NRC fallback), and optionally `afinn` (AFINN fallback).

Packages

```
library(tidyverse)
library(tidytext)
library(janeaustenr)
library(gutenbergr)
library(stopwords)
library(lexicon)
library(lexiconPT)
```

Primary Example from Text Mining with R (Chaper 2): Jane Austen + `bing` / `AFINN` / `nrc`

```
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    # Avoid \d escape; use explicit digit/roman class:
    chapter = cumsum(str_detect(text, regex("^chapter [0-9ivxlc]+", ignore_case = TRUE)))
  ) %>%
```

```
  ungroup() %>%
  unnest_tokens(word, text)

nrc_try <- try(tidytext::get_sentiments("nrc"), silent = TRUE)
```

```
## Do you want to download:
##  Name: NRC Word-Emotion Association Lexicon
##  URL: http://saifmohammad.com/WebPages/lexicons.html
##  License: License required for commercial use. Please contact Saif M. Mohammad (saif.mohammad@nrc-cn
##  Size: 22.8 MB (cleaned 424 KB)
##  Download mechanism: http
##  Citation info:
##
## This dataset was published in Saif M. Mohammad and Peter Turney. (2013), ‘‘Crowdsourcing a Word-Emot
##
## article{mohammad13,
## author = {Mohammad, Saif M. and Turney, Peter D.},
## title = {Crowdsourcing a Word-Emotion Association Lexicon},
## journal = {Computational Intelligence},
## volume = {29},
## number = {3},
## pages = {436-465},
## doi = {10.1111/j.1467-8640.2012.00460.x},
## url = {https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8640.2012.00460.x},
## eprint = {https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8640.2012.00460.x},
## year = {2013}
## }
## If you use this lexicon, then please cite it.
```

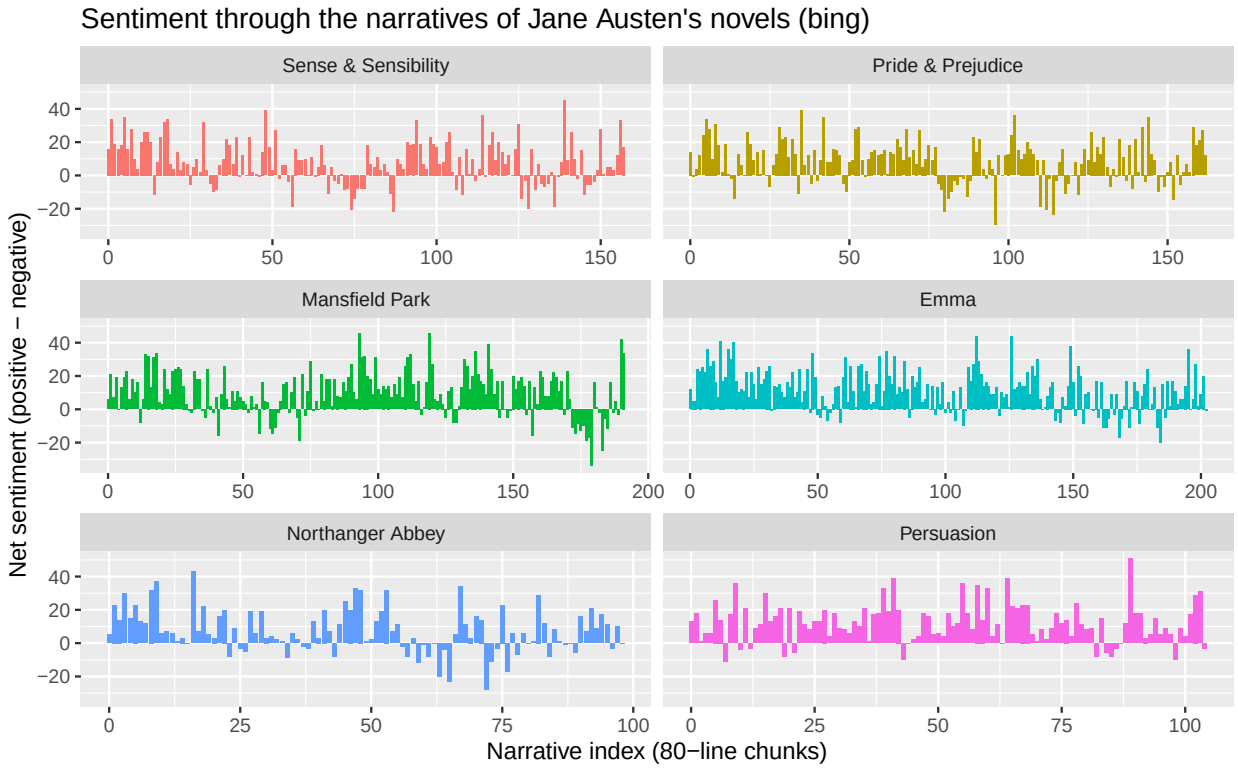
```
if (inherits(nrc_try, "try-error")) {
  nrc_hash <- lexicon::hash_sentiment_nrc
  if (is.data.frame(nrc_hash) && all(c("x","y") %in% names(nrc_hash))) {
    if (is.numeric(nrc_hash$y)) {
      nrc <- tibble(word = nrc_hash$x, sentiment = ifelse(nrc_hash$y > 0, "positive", "negative"))
    } else {
      nrc <- tibble(word = nrc_hash$x, sentiment = as.character(nrc_hash$y))
    }
  } else if (is.list(nrc_hash) || is.environment(nrc_hash)) {
    tmp <- lapply(names(nrc_hash), function(s) tibble(word = nrc_hash[[s]], sentiment = s))
    nrc <- bind_rows(tmp)
  } else {
    nrc <- tibble(word = names(nrc_hash), sentiment = as.character(nrc_hash))
  }
} else {
  nrc <- nrc_try
}

nrc_joy <- nrc %>% filter(sentiment == "joy")

emma_joy_counts <- tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy, by = "word") %>%
  count(word, sort = TRUE)
```

```
head(emma_joy_counts, 15)
```

```
## # A tibble: 0 x 2
## # i 2 variables: word <chr>, n <int>
```

Fig. 2.2 of chapter 2 Sentiment over narrative time with bing

```
bing_try <- try(tidytext::get_sentiments("bing"), silent = TRUE)
if (inherits(bing_try, "try-error")) {
  bl <- lexicon::hash_sentiment_bingliu
  bing_sent <- tibble(
    word = names(bl),
    sentiment = ifelse(as.integer(bl) > 0, "positive", "negative")
  )
} else {
  bing_sent <- bing_try
}

jane_austen_sentiment <- tidy_books %>%
  inner_join(bing_sent, by = "word") %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  tidyr::pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x") +
  labs(
    title = "Sentiment through the narratives of Jane Austen's novels (bing)",
    x = "Narrative index (80-line chunks)",
    y = "Net sentiment (positive - negative)"
  )
```



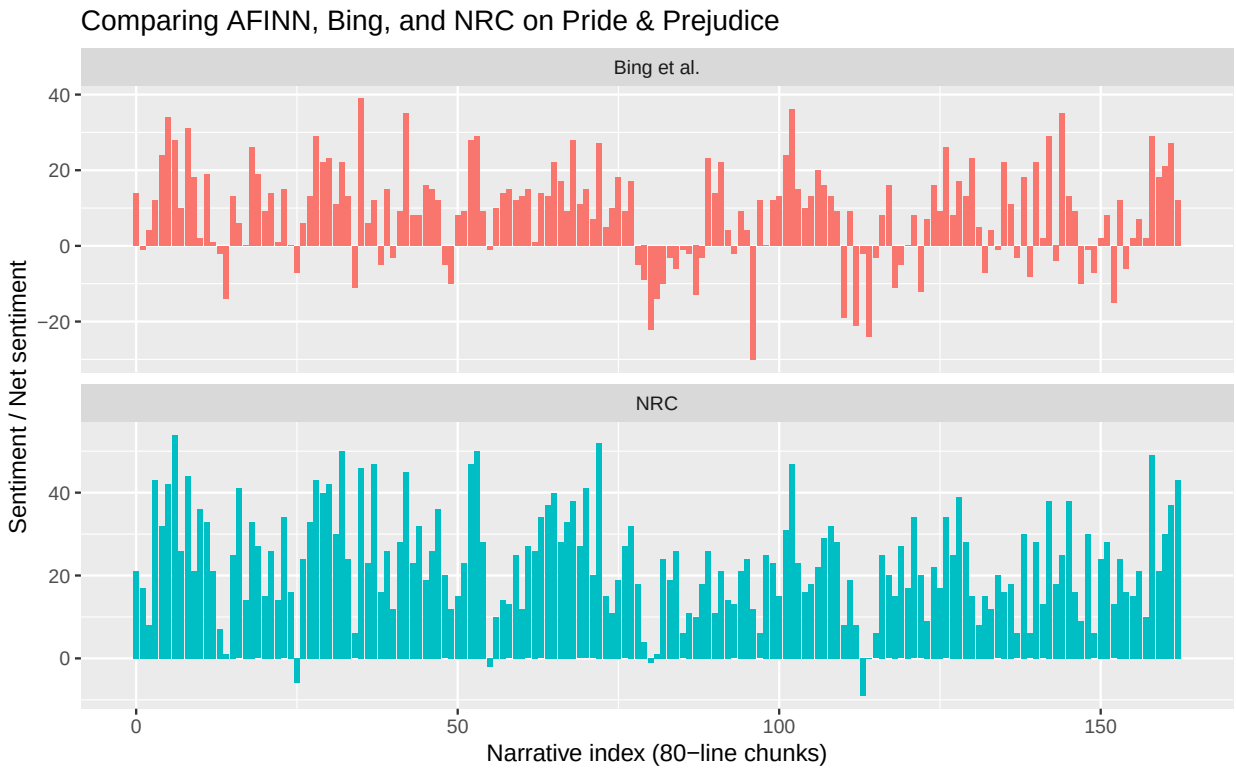
Compare AFINN, bing, NRC on *Pride & Prejudice*

```
pride_prejudice <- tidy_books %>% filter(book == "Pride & Prejudice")

afinn <- NULL
afinn_try <- try(tidytext::get_sentiments("afinn"), silent = TRUE)
```

```
## Do you want to download:
##  Name: AFINN-111
##  URL: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010
##  License: Open Database License (ODbL) v1.0
##  Size: 78 KB (cleaned 59 KB)
##  Download mechanism: https
```

```
if (!inherits(afinn_try, "try-error")) {
  afinn <- afinn_try
} else if (requireNamespace("afinn", quietly = TRUE)) {
  afinn <- afinn::afinn
  if (!all(c("word","value") %in% names(afinn))) afinn <- NULL
}

afinn_df <- NULL
if (!is.null(afinn)) {
  afinn_df <- pride_prejudice %>%
    inner_join(afinn, by = "word") %>%
    group_by(index = linenumber %/% 80) %>%
```

```
    summarise(sentiment = sum(value), .groups = "drop") %>%
    mutate(method = "AFINN")
}

bing_df <- pride_prejudice %>%
  inner_join(bing_sent, by = "word") %>%
  mutate(method = "Bing et al.")

nrc_posneg <- nrc %>% filter(sentiment %in% c("positive","negative"))
nrc_df <- pride_prejudice %>%
  inner_join(nrc_posneg, by = "word") %>%
  mutate(method = "NRC")

bing_and_nrc <- bind_rows(bing_df, nrc_df) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

combo <- if (!is.null(afinn_df)) bind_rows(afinn_df, bing_and_nrc) else bing_and_nrc

ggplot(combo, aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y") +
  labs(
    title = "Comparing AFINN, Bing, and NRC on Pride & Prejudice",
    x = "Narrative index (80-line chunks)",
    y = "Sentiment / Net sentiment"
  )
```



Extension 1 — Portuguese corpus: *Amor de Perdição* (Camilo Castelo Branco)

```
camilo_raw <- gutenberg_download(16425)

camilo <- camilo_raw %>%
  mutate(linenumber = row_number())

tidy_camilo <- camilo %>%
  tidytext::unnest_tokens(word, text, token = "words")

stop_pt <- tibble(word = stopwords::stopwords("pt"))

tidy_camilo <- tidy_camilo %>%
  anti_join(stop_pt, by = "word") %>%
  filter(str_detect(word, "^[[:alpha:]]+$"))

head(tidy_camilo)
```

```
## # A tibble: 6 x 3
##   gutenberg_id linenumber word
##          <int>      <int> <chr>
## 1        16425          1 amor
## 2        16425          1 perdi
## 3        16425          6 amor
## 4        16425          6 perdi
## 5        16425          9 memorias
## 6        16425          9 familia
```

Extension 2 — Additional lexicon: OpLexicon (Portuguese)

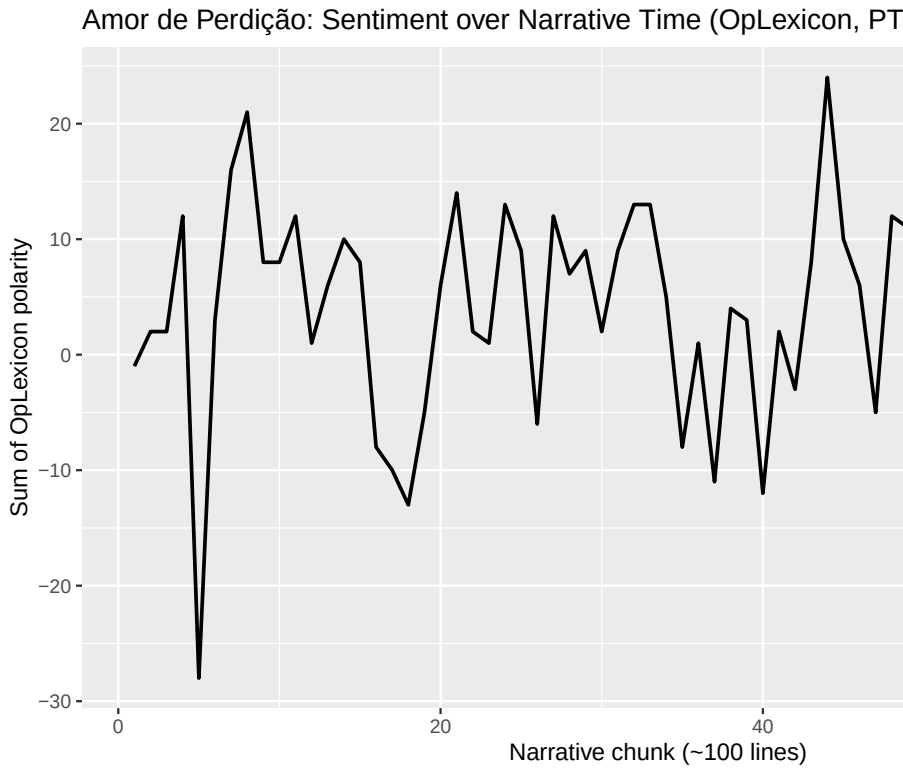
```
data(oplexicon_v3.0, package = "lexiconPT")

pt_lex <- oplexicon_v3.0 %>%
  transmute(word = term, value = as.integer(polarity))

summary(pt_lex$value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0000 -1.0000  0.0000 -0.1848  1.0000  1.0000
```

```
head(pt_lex)
```

```
##   word value
## 1   =[    -1
## 2   =@    -1
## 3   =p    -1
## 4   =P    -1
## 5   =x    -1
## 6   =d     1
```

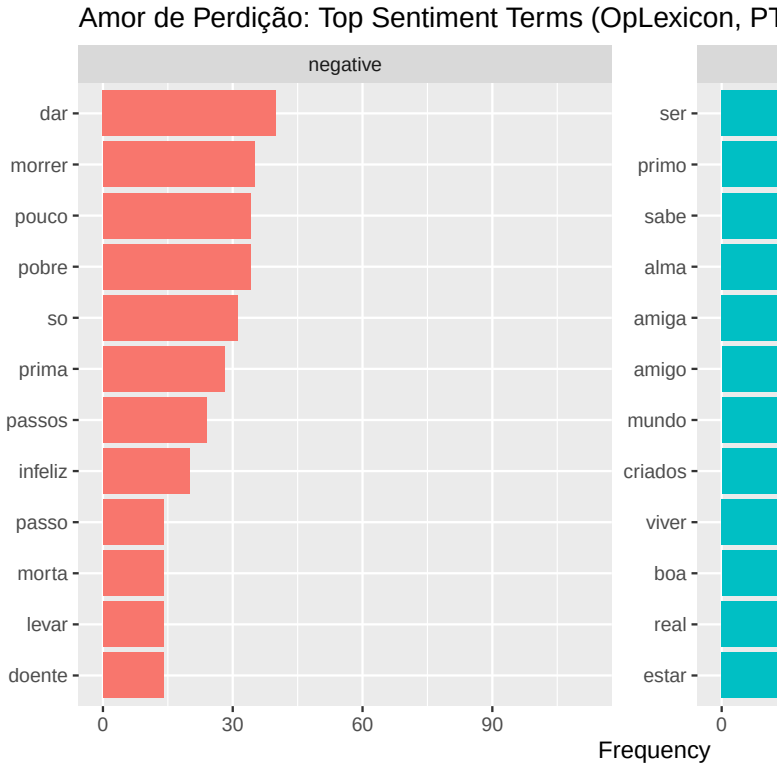
```
camilo_oplex <- tidy_camilo %>%
  inner_join(pt_lex, by = "word") %>%
  mutate(chunk = (linenumber %/% 100) + 1) %>%
  group_by(chunk) %>%
  summarise(score = sum(value), .groups = "drop")

ggplot(camilo_oplex, aes(chunk, score)) +
  geom_line(linewidth = 0.8) +
  labs(
    title = "Amor de Perdição: Sentiment over Narrative Time (OpLexicon, PT)",
    x = "Narrative chunk (~100 lines)",
    y = "Sum of OpLexicon polarity"
  )
```



Narrative-time sentiment (OpLexicon)

```
camilo_words_scored <- tidy_camilo %>%
  inner_join(pt_lex, by = "word")

camilo_posneg <- camilo_words_scored %>%
  mutate(sentiment = case_when(
    value > 0 ~ "positive",
    value < 0 ~ "negative",
    TRUE      ~ "neutral"
  )) %>%
```

```
  filter(sentiment != "neutral") %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  slice_max(n, n = 12) %>%
  ungroup() %>%
  mutate(word = reorder_within(word, n, sentiment))

ggplot(camilo_posneg, aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ sentiment, scales = "free_y") +
  scale_y_reordered() +
  labs(
    title = "Amor de Perdição: Top Sentiment Terms (OpLexicon, PT)",
    x = "Frequency",
    y = NULL
  )
```



Top positive/negative contributors (OpLexicon)

Conclusion

In conclusion, this report closely follows the structure of Chapter 2 of **TextMining with R**, beginning with the identification of NRC "joy" words in Emma, continuing with the analysis of Bing sentiment scores and concluding with a comparison of AFINN, Bing, and NRC sentiment methods in *Pride and Prejudice*. The analysis is then extended to a Portuguese book, *Amor de Perdição* by Camilo Castelo Branco, who is one my favorite Portuguese authors, using the OpLexicon sentiment lexicon to ensure linguistic and cultural accuracy for Portuguese-language data.