DATA 607: Week 10

Cindy Lin

2025-04-01

INTRODUCTION

Week 10's assignment focuses on text mining, and I chose to analyze poetry because it often relies less on individual words and more on overarching themes. Poetry tends to be highly nuanced, with meaning shaped significantly by the reader's interpretation, making it an interesting challenge for text mining techniques. I selected Emily Dickinson's work, as her poetry is frequently noted for its ambiguity and contrasting tones. I'm curious to see how sentiment analysis interprets her writing and whether the results align with the commonly held view of her work as layered and open to multiple interpretations.

I am starting by recreating some of the codes in the Text Mining textbook -

Citation: Silge, J., & Robinson, D. (2017). *Text Mining with R: A Tidy Approach*. O'Reilly Media. https://www.tidytextmining.com/sentiment.html

Loading library

Installing the necessary packages for text mining - gutenbergr package to get Emily Dickinson's work.

Recreating with Emily Dickinson's WorK

```
library(gutenbergr)
library(tidytext)
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(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
library(sentimentr)
library(stringr)
library(knitr)


gutenberg_works(author == "Dickinson, Emily")
```

```
## # A tibble: 4 x 8
##   gutenberg_id title     author gutenberg_author_id language gutenberg_bookshelf
##          <int> <chr>     <chr>                <int> <chr>    <chr>
## 1         2678 Poems by~ Dicki~                 996 en       ""
## 2         2679 Poems by~ Dicki~                 996 en       ""
## 3        12241 Poems by~ Dicki~                 996 en       ""
## 4        12242 Poems by~ Dicki~                 996 en       "Bibliomania"
## # i 2 more variables: rights <chr>, has_text <lgl>
```

```r
emily_poems <- gutenberg_download(12242)
```

```
## Determining mirror for Project Gutenberg from https://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

```r
tidy_emily <- emily_poems %>%
  unnest_tokens(word, text)
# make one word per row

bing_word_counts <- tidy_emily %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment, sort = TRUE) %>%
  ungroup()
```

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

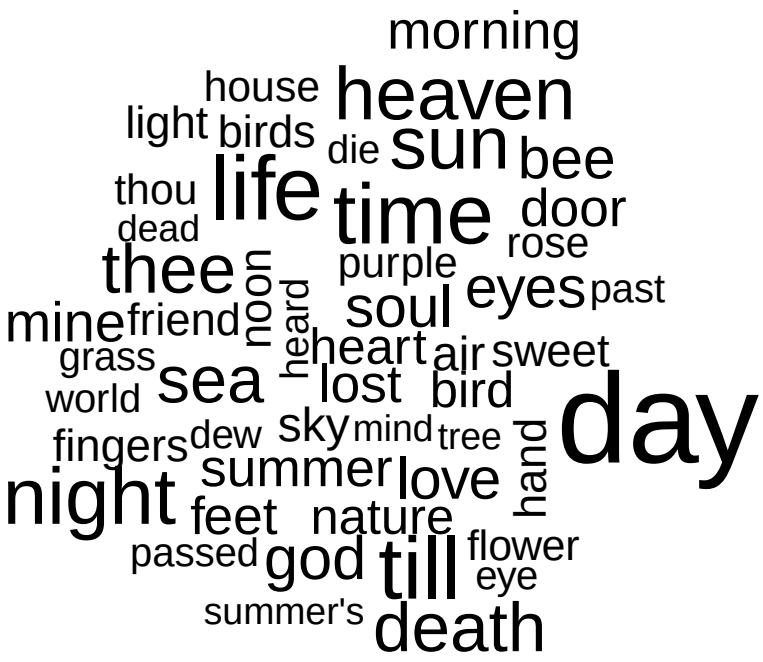
```r
# taken from the text

print(bing_word_counts)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   1231
## 2 positive   1229
```

```r
tidy_emily %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 50))
```

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




```
# word cloud
```

I searched Emily Dickinson's work in the Project Gutenberg collection and downloaded the full series which is the ID: 12242. By joining the token list (one word per row) and matching it to the bing sentiments, it will count up the positive and negative words. Based on the count, there are about the same positive words as negative words in the Emily Dickinson's series of poetry. With words like life, day, night, sun, death, thee, soul, time, heaven, god and mine being the most common words.

Incorporate One Addiitional Sentiment

```
dickinson <- gutenberg_download(12242)


text_combined <- paste(dickinson$text, collapse = " ")
#make into one long text


sentences <- get_sentences(text_combined)
#split into sentences using punctuation rules

flat_sentences <- unlist(sentences)
#make into vector

sentiment_scores <- sentiment(flat_sentences)
```

```
## Warning: Each time ‘sentiment‘ is run it has to do sentence boundary disambiguation when a
## raw ‘character‘ vector is passed to ‘text.var‘. This may be costly of time and
## memory.  It is highly recommended that the user first runs the raw ‘character‘
## vector through the ‘get_sentences‘ function.
```

```
results <- cbind(sentiment_scores, sentence = flat_sentences)
#combine the score and sentences
overall_sentiment <- mean(results$sentiment)

if (overall_sentiment > 0) {
  cat("Overall sentiment is positive.\n")
} else if (overall_sentiment < 0) {
  cat("Overall sentiment is negative.\n")
} else {
  cat("Overall sentiment is neutral.\n")
}
```

```
## Overall sentiment is positive.
```

```
# Get the overall sentiment score

print(overall_sentiment)
```

```
## [1] 0.02373853
```

```
top_pos <- results[order(results$sentiment, decreasing = TRUE), ][1:5, ]
top_neg <- results[order(results$sentiment, decreasing = FALSE), ][1:5, ]




kable(top_pos, format = "latex")
```

element_id	sentence_id	word_count	sentiment	sentence
470	1	1	1.0000000	REFUGE.
1711	1	1	1.0000000	SATISFIED.
17	1	63	0.8630189	It is believed that the thoughtful reader will find in these pages a
1320	1	19	0.8603090	His gait was soundless, like the bird, But rapid, like the roe; His fa
2022	1	22	0.8314828	Please God, might I behold him In epauletted white, I should not

```
kable(top_neg, format = "latex")
```

element_id	sentence_id	word_count	sentiment	sentence
1696	1	2	-1.2374369	CHILDISH GRIEFS.
843	1	1	-1.0000000	REMORSE.
2045	1	1	-1.0000000	INVISIBLE.
2081	1	1	-1.0000000	DEAD.
93	1	14	-0.9621405	Much madness is divinest sense To a discerning eye; Much sense t

I wanted to use Sentimentr because it is a sentence level analysis which would be more appropriate in things like literature and poetry. The overall sentiment score of Emily Dickinson's work is neutral as the score is very close to zero. Because it was netural, I also wanted to do the top 5 postive and negative scores so we can see a general sense of her work from both end of the spectrum.

Using Azure Language Services

```
library(httr)
library(jsonlite)

headers <- add_headers(
  `Content-Type` = "application/json",
  `Ocp-Apim-Subscription-Key` = Sys.getenv("api_key")

)

data <- '{"documents": [{"id": "2", "text": "
If I can stop one heart from breaking,
I shall not live in vain;
If I can ease one life the aching,
Or cool one pain,
Or help one fainting robin
Unto his nest again,
I shall not live in vain."}]}'

res <- POST(
  url = "https://week10data607.cognitiveservices.azure.com/text/analytics/v3.1/sentiment?opinionMining=
  headers,
  body = data,
  encode = "raw"
)

res
```

```
## Response [https://week10data607.cognitiveservices.azure.com/text/analytics/v3.1/sentiment?opinionMini
##   Date: 2025-04-06 21:22
##   Status: 200
##   Content-Type: application/json; charset=utf-8
##   Size: 628 B
```

```
content(res)$documents[[1]]$sentiment
```

```
## [1] "mixed"
```

I am using the Azure Language Services and used a poem that I would imagine to be ambiguous in the individual words but the theme of the poem is inspirational and about compassion. This aligns with my thinking as the sentiment score is mixed.

CONCLUSION

By recreating the word analysis in Text Mining with R, we can better understand the differences in sentiment across various works. This comparison is particularly interesting because while the example in the text focused on literature, the text used here is poetry. Literature often invites a "read between the lines" interpretation, whereas poetry tends to be even more elusive and layered in meaning. By using sentimentr, we went beyond individual word sentiment analysis and explored the overall tone of the text. As expected, the sentiment analysis returned a neutral score, highlighting the subtlety and complexity of poetic expression.