01 The tidy text format

H. David Shea

29 Jul 2021

Contents

The `unnest_tokens` function

```
text <- c(
    "Because I could not stop for Death -",
    "He kindly stopped for me -",
    "The Carriage held but just Ourselves -",
    "and Immortality"
)

kable(text)
```

x
Because I could not stop for Death -
He kindly stopped for me -
The Carriage held but just Ourselves -
and Immortality

```
text_df <- tibble(line = 1:4, text = text)

kable(text_df)
```

line	text
1	Because I could not stop for Death -
2	He kindly stopped for me -
3	The Carriage held but just Ourselves -
4	and Immortality

```
text_df %>%
    unnest_tokens(word, text) %>%
    kable()
```

line	word
1	because
1	i
1	could
1	not
1	stop
1	for
1	death
2	he
2	kindly
2	stopped
2	for
2	me
3	the
3	carriage
3	held
3	but
3	just
3	ourselves
4	and
4	immortality

```
text_df %>%
    unnest_tokens(word, text, to_lower = FALSE) %>%
    kable()
```

line	word
1	Because
1	I
1	could
1	not
1	stop
1	for
1	Death
2	He
2	kindly
2	stopped
2	for
2	me
3	The
3	Carriage
3	held
3	but
3	just
3	Ourselves

line	word
4	and
4	Immortality

Tidying the works of Jane Austen

Get the text from all Jane Austen books, add fields for line number and chapter number. The line number is obtained by a simple `row_number()` call. The chapter number relies on a `cumsum` of each line that starts with the word 'chapter' followed by a space and then a number or any of the (smaller - i.e., no 'm' - not a lot of 1000 chapter books) Roman numeral letters - neat trick.

NOTE: The `austen_books()` data are in text only format - exactly what we want - so no pre-processing is required.

```
original_books <- austen_books() %>%
    group_by(book) %>%
    mutate(linenumber = row_number(),
           chapter = cumsum(str_detect(
               text,
               regex("^chapter [\\divxlc]",
                     ignore_case = TRUE)
           ))) %>%
    ungroup()

kable(original_books[1:10, ])
```

text	book	linenumber	chapter
SENSE AND SENSIBILITY	Sense & Sensibility	1	0
	Sense & Sensibility	2	0
by Jane Austen	Sense & Sensibility	3	0
	Sense & Sensibility	4	0
(1811)	Sense & Sensibility	5	0
	Sense & Sensibility	6	0
	Sense & Sensibility	7	0
	Sense & Sensibility	8	0
	Sense & Sensibility	9	0
CHAPTER 1	Sense & Sensibility	10	1

```
tidy_books <- original_books %>%
    unnest_tokens(word, text)

kable(tidy_books[1:10, ])
```

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

book	linenumber	chapter	word
Sense & Sensibility	3	0	jane
Sense & Sensibility	3	0	austen
Sense & Sensibility	5	0	1811
Sense & Sensibility	10	1	chapter
Sense & Sensibility	10	1	1
Sense & Sensibility	13	1	the

Stop words are words that are not usually useful for analyses. These are the typically high frequency common words like 'the', 'of', 'to', etc. The package `tidytext` contains a dataset `stop_words` containing several lexicons' versions of stop words.

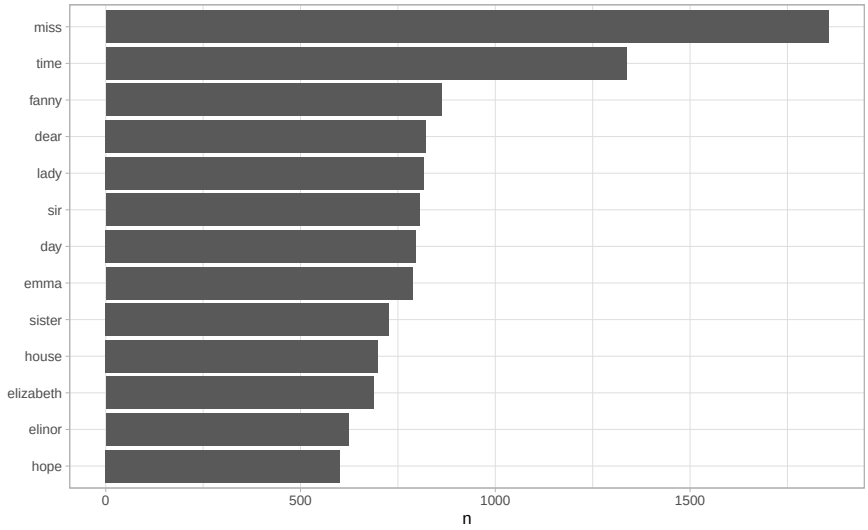
```
data(stop_words)

tidy_books <- tidy_books %>%
    anti_join(stop_words, by = "word")

tidy_books %>%
    count(word, sort = TRUE) %>%
    slice_max(n, n = 10) %>%
    kable()
```

word	n
miss	1855
time	1337
fanny	862
dear	822
lady	817
sir	806
day	797
emma	787
sister	727
house	699

```
tidy_books %>%
    count(word, sort = TRUE) %>%
    filter(n > 600) %>%
    mutate(word = reorder(word, n)) %>%
    ggplot(aes(n, word)) +
    geom_col() +
    labs(y = NULL) +
    theme_light()
```



The `gutenbergr` package

Project Gutenberg is a library of over 60,000 free eBooks. The `gutenbergr` package provides access to these books. Here, we pull the data for some H.G. Wells books: *The Time Machine* (ID = 35), *The War of the Worlds* (ID = 36), *The Invisible Man* (ID = 5230), and *The Island of Doctor Moreau* (ID = 159). Then we do the same for works from the Bronte Sisters: *Jane Eyre* (ID = 1260), *Wuthering Heights* (ID = 768), *The Tenant of Wildfell Hall* (ID = 969), *Villette* (ID = 9182), and *Agnes Grey* (ID = 767).

```
hgwells <- gutenberg_download(c(35, 36, 5230, 159))

tidy_hgwells <- hgwells %>%
    unnest_tokens(word, text) %>%
    anti_join(stop_words, by = "word")

tidy_hgwells %>%
    count(word, sort = TRUE) %>%
    slice_max(n, n = 10) %>%
    kable()
```

word	n
time	461
people	302
door	260
heard	249
black	232
stood	229
white	224
hand	218
kemp	213
eyes	210
suddenly	210

```
bronte <- gutenberg_download(c(1260, 768, 969, 9182, 767))

tidy_bronte <- bronte %>%
    unnest_tokens(word, text) %>%
    anti_join(stop_words, by = "word")

tidy_bronte %>%
    count(word, sort = TRUE) %>%
    slice_max(n, n = 10) %>%
    kable()
```

word	n
time	1064
miss	854
day	826
hand	767
eyes	713
don't	666
night	648
heart	638
looked	601
door	591

Word frequencies

Now we calculate the frequency for each word for the collected work of the set of authors: Jane Austen, the Bronte sisters, and H.G. Wells. This makes good use of `tidverse` operations.

NOTE: The Project Gutenberg books have some examples of emphasized words indicated by underscores. The `str_extract` below, makes sure that only letters and apostrophes are sampled, not the special characters.

```
frequency <-
    bind_rows(
        mutate(tidy_bronte, author = "Brontë Sisters"),
        mutate(tidy_hgwells, author = "H.G. Wells"),
        mutate(tidy_books, author = "Jane Austen")
    ) %>%
    mutate(word = str_extract(word, "[a-z']+")) %>%
    count(author, word) %>%
    group_by(author) %>%
    mutate(proportion = n / sum(n)) %>%
    select(-n) %>%
    pivot_wider(names_from = author, values_from = proportion) %>%
    pivot_longer(`Brontë Sisters`:`H.G. Wells`,
                 names_to = "author",
                 values_to = "proportion")

kable(frequency[1:10, ])
```

word	Jane Austen	author	proportion
a	9.2e-06	Brontë Sisters	0.0000587
a	9.2e-06	H.G. Wells	0.0000148
aback	NA	Brontë Sisters	0.0000039
aback	NA	H.G. Wells	0.0000148
abaht	NA	Brontë Sisters	0.0000039
abaht	NA	H.G. Wells	NA
abandon	NA	Brontë Sisters	0.0000313
abandon	NA	H.G. Wells	0.0000148
abandoned	4.6e-06	Brontë Sisters	0.0000900
abandoned	4.6e-06	H.G. Wells	0.0001778

And this can be used to make a frequency scatter plot to show words used at similar frequencies by the authors - words closer to the abline are similar in frequency.

```
ggplot(frequency, aes(
    x = proportion,
    y = `Jane Austen`,
    color = abs(`Jane Austen` - proportion)
)) +
    geom_abline(color = "gray40", lty = 2) +
    geom_jitter(
        alpha = 0.1,
        size = 2.5,
        width = 0.3,
        height = 0.3,
        na.rm = TRUE
    ) +
    geom_text(
        aes(label = word),
        check_overlap = TRUE,
        vjust = 1.5,
        na.rm = TRUE
    ) +
    scale_x_log10(labels = percent_format()) +
    scale_y_log10(labels = percent_format()) +
    scale_color_gradient(limits = c(0, 0.001),
                         low = "darkslategray4",
                         high = "gray75") +
    facet_wrap( ~ author, ncol = 2) +
    labs(y = "Jane Austen", x = NULL) +
    theme_light() +
    theme(legend.position = "none")
```

Note the difference in shape between the two plots. The Austen-Bronte plots shows more data points, points that are generally closer to the abline and more lower frequency words in common versus the Austen-Wells plot. This indicates that Jane Austen and the Bronte sisters used more similar words than Jane Austen and H.G. Wells did.

This can be shown in correlation tests as well.

```
cor.test(data = frequency[frequency$author == "Brontë Sisters", ],
         ~ proportion + `Jane Austen`)
#>
#>  Pearson's product-moment correlation
#>
#> data:  proportion and Jane Austen
#> t = 111.09, df = 10345, p-value < 2.2e-16
#> alternative hypothesis: true correlation is not equal to 0
#> 95 percent confidence interval:
#>  0.7286568 0.7462330
#> sample estimates:
#>       cor
#> 0.7375698

cor.test(data = frequency[frequency$author == "H.G. Wells", ],
         ~ proportion + `Jane Austen`)
#>
#>  Pearson's product-moment correlation
#>
#> data:  proportion and Jane Austen
#> t = 36.083, df = 6046, p-value < 2.2e-16
#> alternative hypothesis: true correlation is not equal to 0
#> 95 percent confidence interval:
#>  0.3999815 0.4414612
```

```
#> sample estimates:
#>       cor
#> 0.4209414
```