Text Mining

THE TIDY WAY

Julia Silge
@juliasilge
http://juliasilge.com/
TIDY DATA PRINCIPLES CAN MAKE TEXT MINING EASIER AND MORE EFFECTIVE
tidytext: Text mining using dplyr, ggplot2, and other tidy tools

Authors: Julia Silge, David Robinson
License: MIT

Using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use. Much of the infrastructure needed for text mining with tidy data frames already exists in packages like dplyr, broom, tidyr and ggplot2. In this package, we provide functions and supporting data sets to allow conversion of text to and from tidy formats, and to switch seamlessly between tidy tools and existing text mining packages.

https://github.com/juliasilge/tidytext
What do we mean by tidy text?
What do we mean by tidy text?

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

## [1] "Because I could not stop for Death -"   "He kindly stopped for me -"
## [3] "The Carriage held but just Ourselves -" "and Immortality"
What do we mean by tidy text?

```r
> library(dplyr)
> text_df <- data_frame(line = 1:4, text = text)

> text_df

## # A tibble: 4 × 2
## #  line     text
##  <int>   <chr>
## 1     1   Because I could not stop for Death -
## 2     2             He kindly stopped for me -
## 3     3 The Carriage held but just Ourselves -
## 4     4                        and Immortality
```
What do we mean by tidy text?

```r
library(dplyr)

# A tibble: 4 × 2
#  line       text
#  <int>      <chr>
#1      1  Because I could not stop for Death -
#2      2  He kindly stopped for me -
#3      3  The Carriage held but just Ourselves -
#4      4  and Immortality

> library(tidytext)

> text_df %>%
  unnest_tokens(word, text)

# A tibble: 20 × 2
#  line word
#   <int> <chr>
#1      1 because
#2      1 i
#3      1 could
#4      1 not
#5      1 stop
#6      1 for
#7      1 death
#8      2 he
#9      2 kindly
#10     2 stopped
# # ... with 10 more rows
```
What do we mean by tidy text?

> library(tidytext)
> text_df %>%
>   unnest_tokens(word, text)

```r
## # A tibble: 20 × 2
##   line    word
##    <int>   <chr>
##  1      1 because
##  2      1       i
##  3      1   could
##  4      1     not
##  5      1    stop
##  6      1     for
##  7      2      he
##  8      2  kindly
##  9      2 stopped
## 10     2     ... with 10 more rows
```

- Other columns have been retained
- Punctuation has been stripped
- Words have been converted to lowercase
Tidying the works of Jane Austen

```r
> library(janeaustenr)
> library(dplyr)
> library(stringr)
>
> original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text,
                                     regex("^chapter [\d\w]+",
                                          ignore_case = TRUE)))) %>%
  ungroup()
```
Tidying the works of Jane Austen

> original_books
# A tibble: 73,422 × 4

<table>
<thead>
<tr>
<th>text</th>
<th>book</th>
<th>linenumber</th>
<th>chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSE AND SENSIBILITY Sense &amp; Sensibility</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>by Jane Austen Sense &amp; Sensibility</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(1811) Sense &amp; Sensibility</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sense &amp; Sensibility</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CHAPTER 1 Sense &amp; Sensibility</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
# ... with 73,412 more rows
Tidying the works of Jane Austen

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

# A tibble: 725,054 × 4

  book            linenumber chapter word
  <fctr>          <int>   <int>       <chr>
1 Sense & Sensibility          1       0       sense
2 Sense & Sensibility          1       0         and
3 Sense & Sensibility          1       0 sensibility
4 Sense & Sensibility          3       0          by
5 Sense & Sensibility          3       0        jane
6 Sense & Sensibility          3       0      austen
7 Sense & Sensibility          5       0        1811
8 Sense & Sensibility         10       1     chapter
9 Sense & Sensibility         10       1           1
10 Sense & Sensibility         13       1        the

# ... with 725,044 more rows
```
OUR TEXT IS TIDY NOW

WHAT NEXT?
REMOVING STOP WORDS

```r
> data(stop_words)
>
> tidy_books <- tidy_books %>%
  anti_join(stop_words)
>
> tidy_books %>%
  count(word, sort = TRUE)
```
Which States Are Mentioned Most in Song Lyrics?

States like California are mentioned most often

from the Washington Post's Wonkblog
Which States Are Mentioned Most in Song Lyrics?

States like Hawaii and Montana are mentioned more often relative to their population

from the Washington Post’s Wonkblog
Sentiment analysis

```r
> get_sentiments("afinn")
# A tibble: 2,476 × 2
  word         score
  <chr>       <int>
1 abandon     -2
2 abandoned   -2
3 abandons   -2
4 abducted    -2
5 abduction   -2
6 abductions -2
7 abhor      -3
8 abhorred   -3
9 abhorrent  -3
10 abhors    -3
# ... with 2,466 more rows

> get_sentiments("bing")
# A tibble: 6,788 × 2
  word          sentiment
  <chr>          <chr>
1 2-faced      negative
2 2-faces      negative
3 a+           positive
4 abnormal     negative
5 abolish      negative
6 abominable   negative
7 abominably   negative
8 abominate    negative
9 abomination negative
10 abort       negative
# ... with 6,778 more rows

> get_sentiments("nrc")
# A tibble: 13,901 × 2
  word          sentiment
  <chr>          <chr>
1 abacus        trust
2 abandon       fear
3 abandon      negative
4 abandon      sadness
5 abandoned     anger
6 abandoned     fear
7 abandoned     sadness
8 abandonment   anger
9 abandonment   fear
10 abandoned    anger
11 abandon      sadness
12 abandonment  fear
# ... with 13,891 more rows
```
Sentiment analysis

> library(tidyrr)
>
> janeaustensentiment <- tidy_books %>%
inner_join(get_sentiments("bing")) %>%
count(book, index = linenumber %/% 100, sentiment) %>%
spread(sentiment, n, fill = 0) %>%
mutate(sentiment = positive - negative)
Sentiment analysis

> bing_word_counts <- austen_books() %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

Which words contribute to each sentiment?
Sentiment analysis

> bing_word_counts
# A tibble: 2,585 × 3

<table>
<thead>
<tr>
<th>word</th>
<th>sentiment</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>miss</td>
<td>negative</td>
<td>1855</td>
</tr>
<tr>
<td>well</td>
<td>positive</td>
<td>1523</td>
</tr>
<tr>
<td>good</td>
<td>positive</td>
<td>1380</td>
</tr>
<tr>
<td>great</td>
<td>positive</td>
<td>981</td>
</tr>
<tr>
<td>like</td>
<td>positive</td>
<td>725</td>
</tr>
<tr>
<td>better</td>
<td>positive</td>
<td>639</td>
</tr>
<tr>
<td>enough</td>
<td>positive</td>
<td>613</td>
</tr>
<tr>
<td>happy</td>
<td>positive</td>
<td>534</td>
</tr>
<tr>
<td>love</td>
<td>positive</td>
<td>495</td>
</tr>
<tr>
<td>pleasure</td>
<td>positive</td>
<td>462</td>
</tr>
</tbody>
</table>

# ... with 2,575 more rows

Which words contribute to each sentiment?
What is a document about?

TERM FREQUENCY

INVERSE DOCUMENT FREQUENCY

\[ idf(\text{term}) = \ln \left( \frac{n_{\text{documents}}}{n_{\text{documents containing term}}} \right) \]
> book_words <- austen_books() %>%
    unnest_tokens(word, text) %>%
    count(book, word, sort = TRUE) %>%
    ungroup()
>
> total_words <- book_words %>%
    group_by(book) %>%
    summarize(total = sum(n))
>
> book_words <- left_join(book_words, total_words)
## TF-IDF

```r
> book_words
# A tibble: 40,379 × 4

  book       word n total
  <fct>     <chr> <int> <int>
1 Mansfield Park the  6206 160460
2 Mansfield Park to  5475 160460
3 Mansfield Park and 5438 160460
4 Emma       to   5239 160996
5 Emma       the  5201 160996
6 Emma       and  4896 160996
7 Mansfield Park of  4778 160460
8 Pride & Prejudice the 4331 122204
9 Emma       of   4291 160996
10 Pride & Prejudice to  4162 122204

# ... with 40,369 more rows
```
Term Frequency Distribution in Jane Austen's Novels

- Sense & Sensibility
- Pride & Prejudice
- Mansfield Park
- Emma
- Northanger Abbey
- Persuasion
TF-IDF

```r
> book_words <- book_words %>%
  bind_tf_idf(word, book, n)
> book_words

# A tibble: 40,379 × 7

  book   word     n total         tf   idf tf_idf
  <fctr> <chr> <int> <int>      <dbl> <dbl>  <dbl>
1 Mansfield Park   the  6206 160460 0.03867631     0      0
2 Mansfield Park    to  5475 160460 0.03412065     0      0
3 Mansfield Park   and  5438 160460 0.03389007     0      0
4         Emma    to  5239 160996 0.03254118     0      0
5         Emma   the  5201 160996 0.03230515     0      0
6         Emma   and  4896 160996 0.03041069     0      0
7 Mansfield Park    of  4778 160460 0.02977689     0      0
8  Pride & Prejudice   the  4331 122204 0.03544074     0      0
9         Emma   of  4291 160996 0.02665284     0      0
10  Pride & Prejudice    to  4162 122204 0.03405780     0      0
# ... with 40,369 more rows
```
TF-IDF

```r
> book_words %>%
+   select(-total) %>%
+   arrange(desc(tf_idf))
# A tibble: 40,379 x 6
  book                      word     n          tf      idf      tf_idf
  <fctr>            <chr> <int>       <dbl>    <dbl>       <dbl>
1 Sense & Sensibility  elinor   623 0.005193528 1.791759 0.009305552
2 Sense & Sensibility  marianne 492 0.004101470 1.791759 0.007348847
3 Mansfield Park       crawford 493 0.003072417 1.791759 0.005505032
4 Pride & Prejudice    darcy   373 0.003052273 1.791759 0.005468939
5 Persuasion           elliot   254 0.003036207 1.791759 0.005440153
6 Emma                  emma   786 0.004882109 1.098612 0.005363545
7 Northanger Abbey     tilney   196 0.002519928 1.791759 0.004515105
8 Emma                  weston  389 0.002416209 1.791759 0.004329266
9 Pride & Prejudice    bennet   294 0.002405813 1.791759 0.004310639
10 Persuasion          wentworth 191 0.002283132 1.791759 0.004090824
# ... with 40,369 more rows
```
Highest tf-idf words in Jane Austen's Novels

Sense & Sensibility
- elinor
- marianne
- dashwood
- jennings
- willoughby
- brandon
- ferrars
- lucy
- barton
- middleton

Pride & Prejudice
- darcy
- bennet
- bingley
- elizabeth
- wickham
- collins
- lydia
- lizzy
- longbourn
- gardiner

Mansfield Park
- crawford
- edmund
- fanny
- bertram
- norris
- rushworth
- mansfield
- thomas
- julia
- crawford's

Emma
- emma
- weston
- knightley
- elton
- woodhouse
- fairfax
- churchill
- harriet
- hartfield
- bates

Northanger Abbey
- tilney
- catherine
- thorpe
- morland
- allen
- isabella
- eleanor
- northanger
- catherine's
- fullerton

Persuasion
- elliot
- wentworth
- walter
- russell
- musgrove
- uppercross
- kellynch
- lyme
- benwick
- henrietta
• As part of the NASA Datanaunts program, I am working on a project to understand NASA datasets
• Metadata includes title, description, keywords, etc
Highest tf-idf words in NASA Metadata Description Fields

Distribution of tf-idf for words from datasets labeled with select keywords

**ASTROPHYSICS**
- circulars
- gcn
- gbm
- issued
- trajectory
- galleries
- studio
- download
- display
- triggers

**BUDGET**
- omb
- 2016
- fiscal
- financial
- 2015
- strategic
- avoidances
- ofcio
- 2014
- plans

**CLOUDS**
- estimates
- ncdc
- fife
- tovs
- received
- cloud
- meteorology
- 1989
- abstract
- toa

**HUMAN HEALTH**
- wellbeing
- china's
- texts
- vulnerability
- 1950
- children
- china
- esi
- births
- treaty

**SEISMOLOGY**
- risk
- earthquake
- multihazard
- gdp
- acceleration
- category
- pga
- hazard
- hazards
- loss

**SOLAR ACTIVITY**
- acrim
- acrimsat
- shutter
- diode
- sim
- solstice
- irradiance
- tsi
- tce
- sor3solmuvd

NASA metadata from https://data.nasa.gov/data.json
n-grams, networks, & negation
Words paired with 'he' and 'she' in Jane Austen's novels

Women remember, read, and feel while men stop, take, and reply

- remembered
- read
- felt
- resolved
- longed
- heard
- dared
- cried
- feared
- instantly
- expected
- received
- saw
- ran
- caught
- took
- can
- wants
- expressed
- came
- shook
- certainly
- stopt
- knows
- seemed
- married
- comes
- replied
- takes
- stopped

Relative appearance after 'she' compared to 'he'
TAKING TIDY TEXT TO THE NEXT LEVEL

tidying
&
casting
Thank You

Julia Silge
@juliasilge
http://juliasilge.com/