Stat. 653 Homework 3

Jiayi Liu

April 15, 2019

# Read: Chapter 3 and Chapter 4

# Problems:

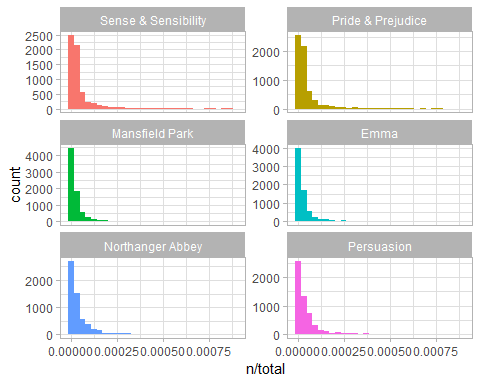
# Run the R code from Chapter 3. 03-tf-idf.Rmd

library(knitr)  
opts\_chunk$set(message = FALSE, warning = FALSE, cache = TRUE)  
options(width = 100, dplyr.width = 100)  
library(ggplot2)  
theme\_set(theme\_light())

library(dplyr)  
library(janeaustenr)  
library(tidytext)  
  
book\_words <- austen\_books() %>%  
 unnest\_tokens(word, text) %>%  
 count(book, word, sort = TRUE)  
  
total\_words <- book\_words %>%   
 group\_by(book) %>%   
 summarize(total = sum(n))  
  
book\_words <- left\_join(book\_words, total\_words)  
  
book\_words

## # A tibble: 40,379 x 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

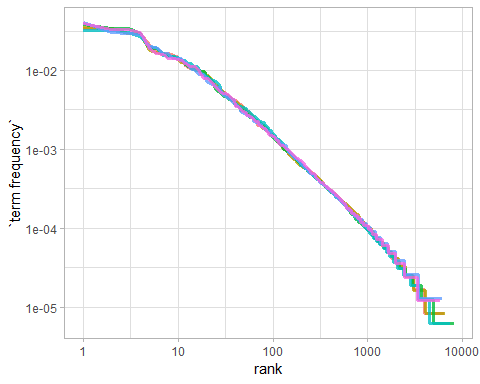
library(ggplot2)  
  
ggplot(book\_words, aes(n/total, fill = book)) +  
 geom\_histogram(show.legend = FALSE) +  
 xlim(NA, 0.0009) +  
 facet\_wrap(~book, ncol = 2, scales = "free\_y")



freq\_by\_rank <- book\_words %>%   
 group\_by(book) %>%   
 mutate(rank = row\_number(),   
 `term frequency` = n/total)  
  
freq\_by\_rank

## # A tibble: 40,379 x 6  
## # Groups: book [6]  
## book word n total rank `term frequency`  
## <fct> <chr> <int> <int> <int> <dbl>  
## 1 Mansfield Park the 6206 160460 1 0.0387  
## 2 Mansfield Park to 5475 160460 2 0.0341  
## 3 Mansfield Park and 5438 160460 3 0.0339  
## 4 Emma to 5239 160996 1 0.0325  
## 5 Emma the 5201 160996 2 0.0323  
## 6 Emma and 4896 160996 3 0.0304  
## 7 Mansfield Park of 4778 160460 4 0.0298  
## 8 Pride & Prejudice the 4331 122204 1 0.0354  
## 9 Emma of 4291 160996 4 0.0267  
## 10 Pride & Prejudice to 4162 122204 2 0.0341  
## # ... with 40,369 more rows

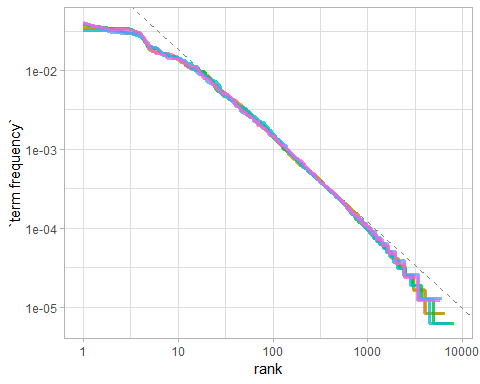
freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = book)) +   
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



rank\_subset <- freq\_by\_rank %>%   
 filter(rank < 500,  
 rank > 10)  
  
lm(log10(`term frequency`) ~ log10(rank), data = rank\_subset)

##   
## Call:  
## lm(formula = log10(`term frequency`) ~ log10(rank), data = rank\_subset)  
##   
## Coefficients:  
## (Intercept) log10(rank)   
## -0.6226 -1.1125

freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = book)) +   
 geom\_abline(intercept = -0.62, slope = -1.1, color = "gray50", linetype = 2) +  
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



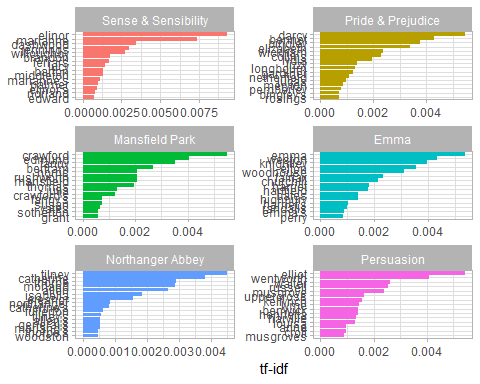
book\_words <- book\_words %>%  
 bind\_tf\_idf(word, book, n)  
book\_words

## # A tibble: 40,379 x 7  
## book word n total tf idf tf\_idf  
## <fct> <chr> <int> <int> <dbl> <dbl> <dbl>  
## 1 Mansfield Park the 6206 160460 0.0387 0 0  
## 2 Mansfield Park to 5475 160460 0.0341 0 0  
## 3 Mansfield Park and 5438 160460 0.0339 0 0  
## 4 Emma to 5239 160996 0.0325 0 0  
## 5 Emma the 5201 160996 0.0323 0 0  
## 6 Emma and 4896 160996 0.0304 0 0  
## 7 Mansfield Park of 4778 160460 0.0298 0 0  
## 8 Pride & Prejudice the 4331 122204 0.0354 0 0  
## 9 Emma of 4291 160996 0.0267 0 0  
## 10 Pride & Prejudice to 4162 122204 0.0341 0 0  
## # ... with 40,369 more rows

book\_words %>%  
 select(-total) %>%  
 arrange(desc(tf\_idf))

## # A tibble: 40,379 x 6  
## book word n tf idf tf\_idf  
## <fct> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Sense & Sensibility elinor 623 0.00519 1.79 0.00931  
## 2 Sense & Sensibility marianne 492 0.00410 1.79 0.00735  
## 3 Mansfield Park crawford 493 0.00307 1.79 0.00551  
## 4 Pride & Prejudice darcy 373 0.00305 1.79 0.00547  
## 5 Persuasion elliot 254 0.00304 1.79 0.00544  
## 6 Emma emma 786 0.00488 1.10 0.00536  
## 7 Northanger Abbey tilney 196 0.00252 1.79 0.00452  
## 8 Emma weston 389 0.00242 1.79 0.00433  
## 9 Pride & Prejudice bennet 294 0.00241 1.79 0.00431  
## 10 Persuasion wentworth 191 0.00228 1.79 0.00409  
## # ... with 40,369 more rows

book\_words %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%   
 group\_by(book) %>%   
 top\_n(15) %>%   
 ungroup() %>%  
 ggplot(aes(word, tf\_idf, fill = book)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~book, ncol = 2, scales = "free") +  
 coord\_flip()

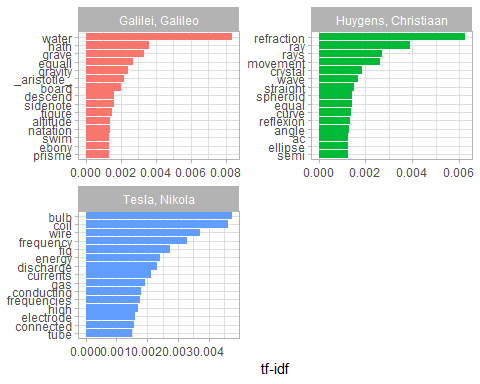


library(gutenbergr)  
physics <- gutenberg\_download(c(37729, 14725, 13476, 1184),   
 meta\_fields = "author")

#load("data/physics.rda")  
physics\_words <- physics %>%  
 unnest\_tokens(word, text) %>%  
 count(author, word, sort = TRUE)  
  
physics\_words

## # A tibble: 9,673 x 3  
## author word n  
## <chr> <chr> <int>  
## 1 Galilei, Galileo the 3760  
## 2 Tesla, Nikola the 3604  
## 3 Huygens, Christiaan the 3553  
## 4 Galilei, Galileo of 2049  
## 5 Tesla, Nikola of 1737  
## 6 Huygens, Christiaan of 1708  
## 7 Huygens, Christiaan to 1207  
## 8 Tesla, Nikola a 1176  
## 9 Galilei, Galileo and 1148  
## 10 Galilei, Galileo to 1133  
## # ... with 9,663 more rows

plot\_physics <- physics\_words %>%  
 bind\_tf\_idf(word, author, n) %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%  
 mutate(author = factor(author, levels = c("Galilei, Galileo",  
 "Huygens, Christiaan",   
 "Tesla, Nikola",  
 "Dumas, Alexandre")))  
  
plot\_physics %>%   
 group\_by(author) %>%   
 top\_n(15, tf\_idf) %>%   
 ungroup() %>%  
 mutate(word = reorder(word, tf\_idf)) %>%  
 ggplot(aes(word, tf\_idf, fill = author)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~author, ncol = 2, scales = "free") +  
 coord\_flip()



library(stringr)  
  
physics %>%   
 filter(str\_detect(text, "she\\.")) %>%   
 select(text)

## # A tibble: 25 x 1  
## text   
## <chr>   
## 1 or dead,” said she.   
## 2 “And when did you arrive here?” inquired she.   
## 3 “’Tis indeed they!” said she.   
## 4 “Yes,” said Madame Danglars, “that is she. Shall I tell you what you  
## 5 “Thank you, thank you,” murmured she. The old man’s declaration that  
## 6 “My hand?” said she.   
## 7 was she.   
## 8 “You here at this hour?” said she.   
## 9 “But, in a word, what are you going to do?” asked she.   
## 10 know what you mean to do!” said she. “Where are you going?”   
## # ... with 15 more rows

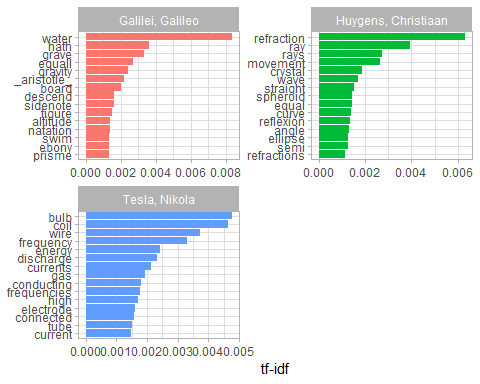
physics %>%   
 filter(str\_detect(text, "yes")) %>%   
 select(text)

## # A tibble: 825 x 1  
## text   
## <chr>   
## 1 eyes, and hair as dark as a raven’s wing; and his whole appearance   
## 2 “Yes—yes: poor Captain Leclere! He was a brave and an honest man.”   
## 3 Son. ‘Ah, yes,’ he said, ‘I know them. The Morrels have been shipowners   
## 4 it will bring tears into the old soldier’s eyes. Come, come,” continued   
## 5 “Yes, yes, I know how good a son you are, so now hasten away to see your  
## 6 “Without her captain!” cried Dantès, his eyes sparkling with animation;   
## 7 “Ah, M. Morrel,” exclaimed the young seaman, with tears in his eyes, and  
## 8 The shipowner, smiling, followed him with his eyes until he saw him   
## 9 “Yes, yes, my boy, so we will—so we will,” replied the old man; “but how  
## 10 “Yes, yes, father, tell me where it is,” and he opened two or three   
## # ... with 815 more rows

physics %>%   
 filter(str\_detect(text, "AK")) %>%   
 select(text)

## # A tibble: 34 x 1  
## text   
## <chr>   
## 1 Now let us assume that the ray has come from A to C along AK, KC; the   
## 2 be equal to the time along KMN. But the time along AK is longer than   
## 3 that along AL: hence the time along AKN is longer than that along ABC.  
## 4 And KC being longer than KN, the time along AKC will exceed, by as   
## 5 line which is comprised between the perpendiculars AK, BL. Then it   
## 6 ordinary refraction. Now it appears that AK and BL dip down toward the  
## 7 side where the air is less easy to penetrate: for AK being longer than  
## 8 than do AK, BL. And this suffices to show that the ray will continue   
## 9 surface AB at the points AK\_k\_B. Then instead of the hemispherical   
## 10 along AL, LB, and along AK, KB, are always represented by the line AH,  
## # ... with 24 more rows

mystopwords <- tibble(word = c("eq", "co", "rc", "ac", "ak", "bn",   
 "fig", "file", "cg", "cb", "cm"))  
physics\_words <- anti\_join(physics\_words, mystopwords, by = "word")  
plot\_physics <- physics\_words %>%  
 bind\_tf\_idf(word, author, n) %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%  
 group\_by(author) %>%   
 top\_n(15, tf\_idf) %>%  
 ungroup() %>%  
 mutate(author = factor(author, levels = c("Galilei, Galileo",  
 "Huygens, Christiaan",  
 "Tesla, Nikola",  
 "Dumas, Alexandre")))  
  
ggplot(plot\_physics, aes(word, tf\_idf, fill = author)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~author, ncol = 2, scales = "free") +  
 coord\_flip()



# Run the code from frequencies.R on the harrypotter data.

library(tidyverse)  
library(harrypotter)  
library(tidytext)  
  
titles <- c("Philosopher's Stone", "Chamber of Secrets", "Prisoner of Azkaban",  
 "Goblet of Fire", "Order of the Phoenix", "Half-Blood Prince",  
 "Deathly Hallows")  
  
books <- list(philosophers\_stone, chamber\_of\_secrets, prisoner\_of\_azkaban,  
 goblet\_of\_fire, order\_of\_the\_phoenix, half\_blood\_prince,  
 deathly\_hallows)  
   
series <- tibble()  
  
for(i in seq\_along(titles)) {  
   
 clean <- tibble(chapter = seq\_along(books[[i]]),  
 text = books[[i]]) %>%  
 unnest\_tokens(word, text) %>%  
 mutate(book = titles[i]) %>%  
 select(book, everything())  
  
 series <- rbind(series, clean)  
}  
  
# set factor to keep books in order of publication  
series$book <- factor(series$book, levels = rev(titles))  
  
series

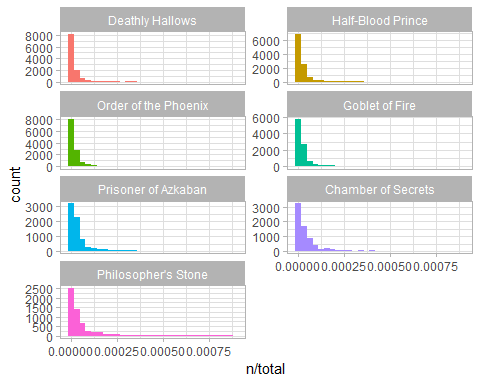
## # A tibble: 1,089,386 x 3  
## book chapter word   
## <fct> <int> <chr>   
## 1 Philosopher's Stone 1 the   
## 2 Philosopher's Stone 1 boy   
## 3 Philosopher's Stone 1 who   
## 4 Philosopher's Stone 1 lived   
## 5 Philosopher's Stone 1 mr   
## 6 Philosopher's Stone 1 and   
## 7 Philosopher's Stone 1 mrs   
## 8 Philosopher's Stone 1 dursley  
## 9 Philosopher's Stone 1 of   
## 10 Philosopher's Stone 1 number   
## # ... with 1,089,376 more rows

book\_words <- series %>%  
 count(book, word, sort = TRUE)  
  
total\_words <- book\_words %>%   
 group\_by(book) %>%   
 summarize(total = sum(n))

book\_words <- left\_join(book\_words, total\_words)  
  
book\_words

## # A tibble: 67,881 x 4  
## book word n total  
## <fct> <chr> <int> <int>  
## 1 Order of the Phoenix the 11740 258763  
## 2 Deathly Hallows the 10335 198906  
## 3 Goblet of Fire the 9305 191882  
## 4 Half-Blood Prince the 7508 171284  
## 5 Order of the Phoenix to 6518 258763  
## 6 Order of the Phoenix and 6189 258763  
## 7 Deathly Hallows and 5510 198906  
## 8 Order of the Phoenix of 5332 258763  
## 9 Prisoner of Azkaban the 4990 105275  
## 10 Goblet of Fire and 4959 191882  
## # ... with 67,871 more rows

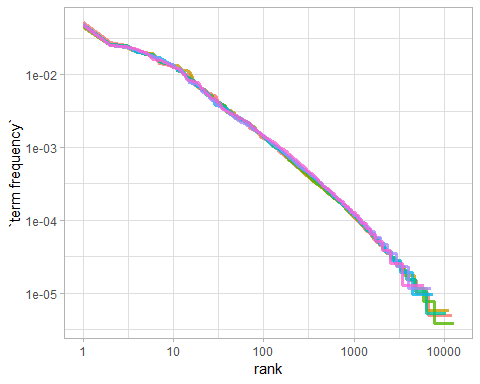
ggplot(book\_words, aes(n/total, fill = book)) +  
 geom\_histogram(show.legend = FALSE) +  
 xlim(NA, 0.0009) +  
 facet\_wrap(~book, ncol = 2, scales = "free\_y")



freq\_by\_rank <- book\_words %>%   
 group\_by(book) %>%   
 mutate(rank = row\_number(),   
 `term frequency` = n/total)  
  
freq\_by\_rank

## # A tibble: 67,881 x 6  
## # Groups: book [7]  
## book word n total rank `term frequency`  
## <fct> <chr> <int> <int> <int> <dbl>  
## 1 Order of the Phoenix the 11740 258763 1 0.0454  
## 2 Deathly Hallows the 10335 198906 1 0.0520  
## 3 Goblet of Fire the 9305 191882 1 0.0485  
## 4 Half-Blood Prince the 7508 171284 1 0.0438  
## 5 Order of the Phoenix to 6518 258763 2 0.0252  
## 6 Order of the Phoenix and 6189 258763 3 0.0239  
## 7 Deathly Hallows and 5510 198906 2 0.0277  
## 8 Order of the Phoenix of 5332 258763 4 0.0206  
## 9 Prisoner of Azkaban the 4990 105275 1 0.0474  
## 10 Goblet of Fire and 4959 191882 2 0.0258  
## # ... with 67,871 more rows

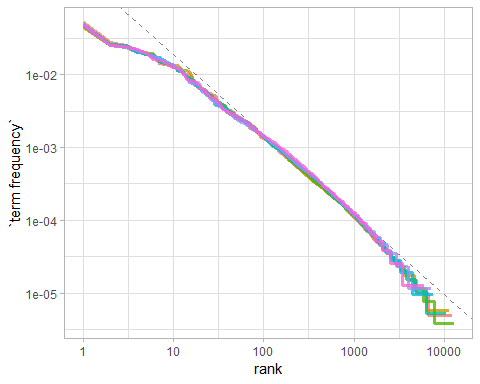
freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = book)) +   
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



rank\_subset <- freq\_by\_rank %>%   
 filter(rank < 500,  
 rank > 10)  
  
lm(log10(`term frequency`) ~ log10(rank), data = rank\_subset)

##   
## Call:  
## lm(formula = log10(`term frequency`) ~ log10(rank), data = rank\_subset)  
##   
## Coefficients:  
## (Intercept) log10(rank)   
## -0.8444 -1.0104

# Classic Zipf's law frequency = \frac{1}{rank}  
  
freq\_by\_rank %>%   
 ggplot(aes(rank, `term frequency`, color = book)) +   
 geom\_abline(intercept = -0.62, slope = -1.1, color = "gray50", linetype = 2) +  
 geom\_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +   
 scale\_x\_log10() +  
 scale\_y\_log10()



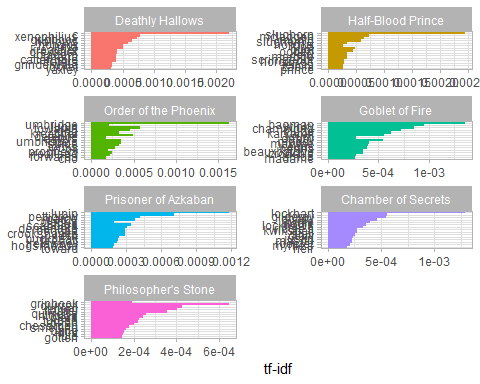
# tf-idf  
  
book\_words <- book\_words %>%  
 bind\_tf\_idf(word, book, n) # idf is the log the result almost 0  
book\_words

## # A tibble: 67,881 x 7  
## book word n total tf idf tf\_idf  
## <fct> <chr> <int> <int> <dbl> <dbl> <dbl>  
## 1 Order of the Phoenix the 11740 258763 0.0454 0 0  
## 2 Deathly Hallows the 10335 198906 0.0520 0 0  
## 3 Goblet of Fire the 9305 191882 0.0485 0 0  
## 4 Half-Blood Prince the 7508 171284 0.0438 0 0  
## 5 Order of the Phoenix to 6518 258763 0.0252 0 0  
## 6 Order of the Phoenix and 6189 258763 0.0239 0 0  
## 7 Deathly Hallows and 5510 198906 0.0277 0 0  
## 8 Order of the Phoenix of 5332 258763 0.0206 0 0  
## 9 Prisoner of Azkaban the 4990 105275 0.0474 0 0  
## 10 Goblet of Fire and 4959 191882 0.0258 0 0  
## # ... with 67,871 more rows

book\_words %>%  
 select(-total) %>%  
 arrange(desc(tf\_idf))

## # A tibble: 67,881 x 6  
## book word n tf idf tf\_idf  
## <fct> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Half-Blood Prince slughorn 335 0.00196 1.25 0.00245   
## 2 Deathly Hallows c 1300 0.00654 0.336 0.00220   
## 3 Order of the Phoenix umbridge 496 0.00192 0.847 0.00162   
## 4 Goblet of Fire bagman 208 0.00108 1.25 0.00136   
## 5 Chamber of Secrets lockhart 197 0.00231 0.560 0.00129   
## 6 Prisoner of Azkaban lupin 369 0.00351 0.336 0.00118   
## 7 Goblet of Fire winky 145 0.000756 1.25 0.000947  
## 8 Goblet of Fire champions 84 0.000438 1.95 0.000852  
## 9 Deathly Hallows xenophilius 79 0.000397 1.95 0.000773  
## 10 Half-Blood Prince mclaggen 65 0.000379 1.95 0.000738  
## # ... with 67,871 more rows

book\_words %>%  
 arrange(desc(tf\_idf)) %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%   
 group\_by(book) %>%   
 top\_n(15) %>%   
 ungroup() %>%  
 ggplot(aes(word, tf\_idf, fill = book)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~book, ncol = 2, scales = "free") +  
 coord\_flip()



# Run the R code from Chapter 4. 04-word-combinations.Rmd Use the R Project I have provided. ngrams.zip Some of the code needs to be changed to make a Word Document.

library(knitr)  
opts\_chunk$set(message = FALSE, warning = FALSE, cache = TRUE)  
options(width = 100, dplyr.width = 100)  
library(ggplot2)  
theme\_set(theme\_light())

library(dplyr)  
library(tidytext)  
library(janeaustenr)  
  
austen\_bigrams <- austen\_books() %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
austen\_bigrams

## # A tibble: 725,049 x 2  
## book bigram   
## <fct> <chr>   
## 1 Sense & Sensibility sense and   
## 2 Sense & Sensibility and sensibility  
## 3 Sense & Sensibility sensibility by   
## 4 Sense & Sensibility by jane   
## 5 Sense & Sensibility jane austen   
## 6 Sense & Sensibility austen 1811   
## 7 Sense & Sensibility 1811 chapter   
## 8 Sense & Sensibility chapter 1   
## 9 Sense & Sensibility 1 the   
## 10 Sense & Sensibility the family   
## # ... with 725,039 more rows

austen\_bigrams %>%  
 count(bigram, sort = TRUE)

## # A tibble: 211,236 x 2  
## bigram n  
## <chr> <int>  
## 1 of the 3017  
## 2 to be 2787  
## 3 in the 2368  
## 4 it was 1781  
## 5 i am 1545  
## 6 she had 1472  
## 7 of her 1445  
## 8 to the 1387  
## 9 she was 1377  
## 10 had been 1299  
## # ... with 211,226 more rows

library(tidyr)  
  
bigrams\_separated <- austen\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_filtered <- bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
  
bigram\_counts

## # A tibble: 33,421 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 sir thomas 287  
## 2 miss crawford 215  
## 3 captain wentworth 170  
## 4 miss woodhouse 162  
## 5 frank churchill 132  
## 6 lady russell 118  
## 7 lady bertram 114  
## 8 sir walter 113  
## 9 miss fairfax 109  
## 10 colonel brandon 108  
## # ... with 33,411 more rows

bigrams\_united <- bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
  
bigrams\_united

## # A tibble: 44,784 x 2  
## book bigram   
## <fct> <chr>   
## 1 Sense & Sensibility jane austen   
## 2 Sense & Sensibility austen 1811   
## 3 Sense & Sensibility 1811 chapter   
## 4 Sense & Sensibility chapter 1   
## 5 Sense & Sensibility norland park   
## 6 Sense & Sensibility surrounding acquaintance  
## 7 Sense & Sensibility late owner   
## 8 Sense & Sensibility advanced age   
## 9 Sense & Sensibility constant companion   
## 10 Sense & Sensibility happened ten   
## # ... with 44,774 more rows

austen\_books() %>%  
 unnest\_tokens(trigram, text, token = "ngrams", n = 3) %>%  
 separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%  
 filter(!word1 %in% stop\_words$word,  
 !word2 %in% stop\_words$word,  
 !word3 %in% stop\_words$word) %>%  
 count(word1, word2, word3, sort = TRUE)

## # A tibble: 8,757 x 4  
## word1 word2 word3 n  
## <chr> <chr> <chr> <int>  
## 1 dear miss woodhouse 23  
## 2 miss de bourgh 18  
## 3 lady catherine de 14  
## 4 catherine de bourgh 13  
## 5 poor miss taylor 11  
## 6 sir walter elliot 11  
## 7 ten thousand pounds 11  
## 8 dear sir thomas 10  
## 9 twenty thousand pounds 8  
## 10 replied miss crawford 7  
## # ... with 8,747 more rows

bigrams\_filtered %>%  
 filter(word2 == "street") %>%  
 count(book, word1, sort = TRUE)

## # A tibble: 34 x 3  
## book word1 n  
## <fct> <chr> <int>  
## 1 Sense & Sensibility berkeley 16  
## 2 Sense & Sensibility harley 16  
## 3 Northanger Abbey pulteney 14  
## 4 Northanger Abbey milsom 11  
## 5 Mansfield Park wimpole 10  
## 6 Pride & Prejudice gracechurch 9  
## 7 Sense & Sensibility conduit 6  
## 8 Sense & Sensibility bond 5  
## 9 Persuasion milsom 5  
## 10 Persuasion rivers 4  
## # ... with 24 more rows

bigram\_tf\_idf <- bigrams\_united %>%  
 count(book, bigram) %>%  
 bind\_tf\_idf(bigram, book, n) %>%  
 arrange(desc(tf\_idf))  
  
bigram\_tf\_idf

## # A tibble: 36,217 x 6  
## book bigram n tf idf tf\_idf  
## <fct> <chr> <int> <dbl> <dbl> <dbl>  
## 1 Persuasion captain wentworth 170 0.0299 1.79 0.0535  
## 2 Mansfield Park sir thomas 287 0.0287 1.79 0.0515  
## 3 Mansfield Park miss crawford 215 0.0215 1.79 0.0386  
## 4 Persuasion lady russell 118 0.0207 1.79 0.0371  
## 5 Persuasion sir walter 113 0.0198 1.79 0.0356  
## 6 Emma miss woodhouse 162 0.0170 1.79 0.0305  
## 7 Northanger Abbey miss tilney 82 0.0159 1.79 0.0286  
## 8 Sense & Sensibility colonel brandon 108 0.0150 1.79 0.0269  
## 9 Emma frank churchill 132 0.0139 1.79 0.0248  
## 10 Pride & Prejudice lady catherine 100 0.0138 1.79 0.0247  
## # ... with 36,207 more rows

library(ggplot2)  
  
bigram\_tf\_idf %>%  
 arrange(desc(tf\_idf)) %>%  
 group\_by(book) %>%  
 top\_n(12, tf\_idf) %>%  
 ungroup() %>%  
 mutate(bigram = reorder(bigram, tf\_idf)) %>%  
 ggplot(aes(bigram, tf\_idf, fill = book)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ book, ncol = 2, scales = "free") +  
 coord\_flip() +  
 labs(y = "tf-idf of bigram to novel",  
 x = "")



bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 count(word1, word2, sort = TRUE)

## # A tibble: 1,246 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 not be 610  
## 2 not to 355  
## 3 not have 327  
## 4 not know 252  
## 5 not a 189  
## 6 not think 176  
## 7 not been 160  
## 8 not the 147  
## 9 not at 129  
## 10 not in 118  
## # ... with 1,236 more rows

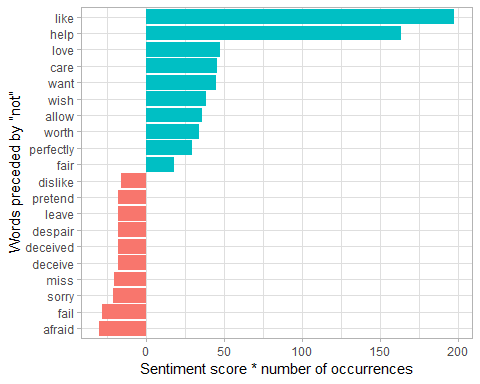
AFINN <- get\_sentiments("afinn")  
  
AFINN

## # A tibble: 2,476 x 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

not\_words <- bigrams\_separated %>%  
 filter(word1 == "not") %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word2, score, sort = TRUE)  
  
not\_words

## # A tibble: 245 x 3  
## word2 score n  
## <chr> <int> <int>  
## 1 like 2 99  
## 2 help 2 82  
## 3 want 1 45  
## 4 wish 1 39  
## 5 allow 1 36  
## 6 care 2 23  
## 7 sorry -1 21  
## 8 leave -1 18  
## 9 pretend -1 18  
## 10 worth 2 17  
## # ... with 235 more rows

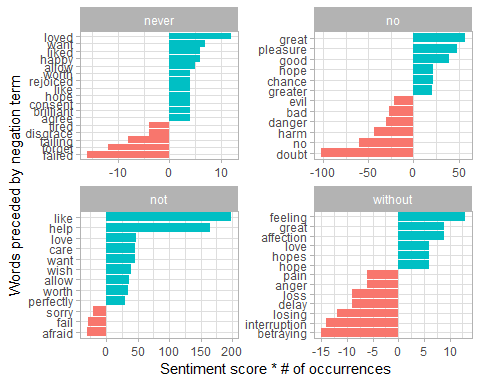
library(ggplot2)  
  
not\_words %>%  
 mutate(contribution = n \* score) %>%  
 arrange(desc(abs(contribution))) %>%  
 head(20) %>%  
 mutate(word2 = reorder(word2, contribution)) %>%  
 ggplot(aes(word2, n \* score, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 xlab("Words preceded by \"not\"") +  
 ylab("Sentiment score \* number of occurrences") +  
 coord\_flip()



negation\_words <- c("not", "no", "never", "without")  
  
negated\_words <- bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word1, word2, score, sort = TRUE)  
negated\_words

## # A tibble: 531 x 4  
## word1 word2 score n  
## <chr> <chr> <int> <int>  
## 1 no doubt -1 102  
## 2 not like 2 99  
## 3 not help 2 82  
## 4 no no -1 60  
## 5 not want 1 45  
## 6 not wish 1 39  
## 7 not allow 1 36  
## 8 not care 2 23  
## 9 no harm -2 22  
## 10 not sorry -1 21  
## # ... with 521 more rows

negated\_words %>%  
 mutate(contribution = n \* score,  
 word2 = reorder(paste(word2, word1, sep = "\_\_"), contribution)) %>%  
 group\_by(word1) %>%  
 top\_n(12, abs(contribution)) %>%  
 ggplot(aes(word2, contribution, fill = n \* score > 0)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ word1, scales = "free") +  
 scale\_x\_discrete(labels = function(x) gsub("\_\_.+$", "", x)) +  
 xlab("Words preceded by negation term") +  
 ylab("Sentiment score \* # of occurrences") +  
 coord\_flip()



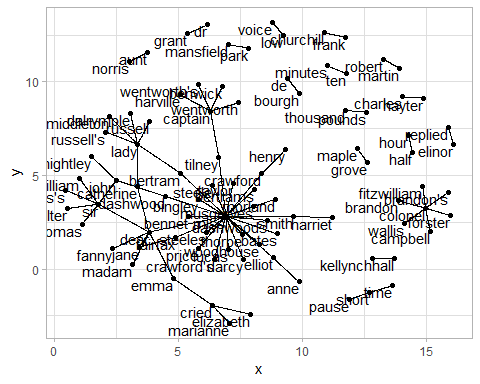
library(igraph)  
  
# original counts  
bigram\_counts

## # A tibble: 33,421 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 sir thomas 287  
## 2 miss crawford 215  
## 3 captain wentworth 170  
## 4 miss woodhouse 162  
## 5 frank churchill 132  
## 6 lady russell 118  
## 7 lady bertram 114  
## 8 sir walter 113  
## 9 miss fairfax 109  
## 10 colonel brandon 108  
## # ... with 33,411 more rows

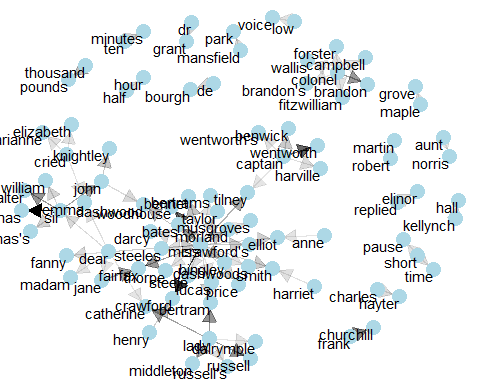
# filter for only relatively common combinations  
bigram\_graph <- bigram\_counts %>%  
 filter(n > 20) %>%  
 graph\_from\_data\_frame()  
  
bigram\_graph

## IGRAPH b00df34 DN-- 91 77 --   
## + attr: name (v/c), n (e/n)  
## + edges from b00df34 (vertex names):  
## [1] sir ->thomas miss ->crawford captain ->wentworth miss ->woodhouse   
## [5] frank ->churchill lady ->russell lady ->bertram sir ->walter   
## [9] miss ->fairfax colonel ->brandon miss ->bates lady ->catherine   
## [13] sir ->john jane ->fairfax miss ->tilney lady ->middleton   
## [17] miss ->bingley thousand->pounds miss ->dashwood miss ->bennet   
## [21] john ->knightley miss ->morland captain ->benwick dear ->miss   
## [25] miss ->smith miss ->crawford's henry ->crawford miss ->elliot   
## [29] dr ->grant miss ->bertram sir ->thomas's ten ->minutes   
## + ... omitted several edges

library(ggraph)  
set.seed(2017)  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)



set.seed(2016)  
  
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE,  
 arrow = a, end\_cap = circle(.07, 'inches')) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()



library(dplyr)  
library(tidyr)  
library(tidytext)  
library(ggplot2)  
library(igraph)  
library(ggraph)  
  
count\_bigrams <- function(dataset) {  
 dataset %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2) %>%  
 separate(bigram, c("word1", "word2"), sep = " ") %>%  
 filter(!word1 %in% stop\_words$word,  
 !word2 %in% stop\_words$word) %>%  
 count(word1, word2, sort = TRUE)  
}  
  
visualize\_bigrams <- function(bigrams) {  
 set.seed(2016)  
 a <- grid::arrow(type = "closed", length = unit(.15, "inches"))  
   
 bigrams %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n), show.legend = FALSE, arrow = a) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()  
}

knitr::include\_graphics("images/widyr.jpg")

austen\_section\_words <- austen\_books() %>%  
 filter(book == "Pride & Prejudice") %>%  
 mutate(section = row\_number() %/% 10) %>%  
 filter(section > 0) %>%  
 unnest\_tokens(word, text) %>%  
 filter(!word %in% stop\_words$word)  
  
austen\_section\_words

## # A tibble: 37,240 x 3  
## book section word   
## <fct> <dbl> <chr>   
## 1 Pride & Prejudice 1 truth   
## 2 Pride & Prejudice 1 universally   
## 3 Pride & Prejudice 1 acknowledged  
## 4 Pride & Prejudice 1 single   
## 5 Pride & Prejudice 1 possession   
## 6 Pride & Prejudice 1 fortune   
## 7 Pride & Prejudice 1 wife   
## 8 Pride & Prejudice 1 feelings   
## 9 Pride & Prejudice 1 views   
## 10 Pride & Prejudice 1 entering   
## # ... with 37,230 more rows

### Counting and correlating among sections

library(widyr)  
  
# count words co-occuring within sections  
word\_pairs <- austen\_section\_words %>%  
 pairwise\_count(word, section, sort = TRUE)  
  
word\_pairs

## # A tibble: 796,008 x 3  
## item1 item2 n  
## <chr> <chr> <dbl>  
## 1 darcy elizabeth 144  
## 2 elizabeth darcy 144  
## 3 miss elizabeth 110  
## 4 elizabeth miss 110  
## 5 elizabeth jane 106  
## 6 jane elizabeth 106  
## 7 miss darcy 92  
## 8 darcy miss 92  
## 9 elizabeth bingley 91  
## 10 bingley elizabeth 91  
## # ... with 795,998 more rows

word\_pairs %>%  
 filter(item1 == "darcy")

## # A tibble: 2,930 x 3  
## item1 item2 n  
## <chr> <chr> <dbl>  
## 1 darcy elizabeth 144  
## 2 darcy miss 92  
## 3 darcy bingley 86  
## 4 darcy jane 46  
## 5 darcy bennet 45  
## 6 darcy sister 45  
## 7 darcy time 41  
## 8 darcy lady 38  
## 9 darcy friend 37  
## 10 darcy wickham 37  
## # ... with 2,920 more rows

### Pairwise correlation

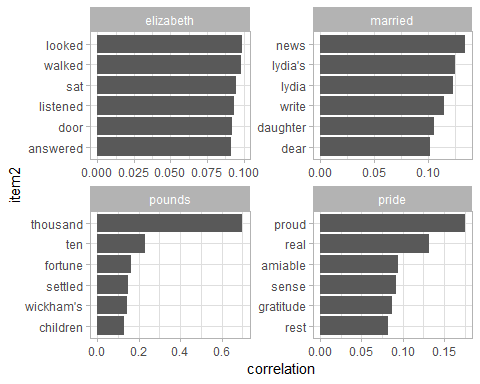
# we need to filter for at least relatively common words first  
word\_cors <- austen\_section\_words %>%  
 group\_by(word) %>%  
 filter(n() >= 20) %>%  
 pairwise\_cor(word, section, sort = TRUE)  
  
word\_cors

## # A tibble: 154,842 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 bourgh de 0.951  
## 2 de bourgh 0.951  
## 3 pounds thousand 0.701  
## 4 thousand pounds 0.701  
## 5 william sir 0.664  
## 6 sir william 0.664  
## 7 catherine lady 0.663  
## 8 lady catherine 0.663  
## 9 forster colonel 0.622  
## 10 colonel forster 0.622  
## # ... with 154,832 more rows

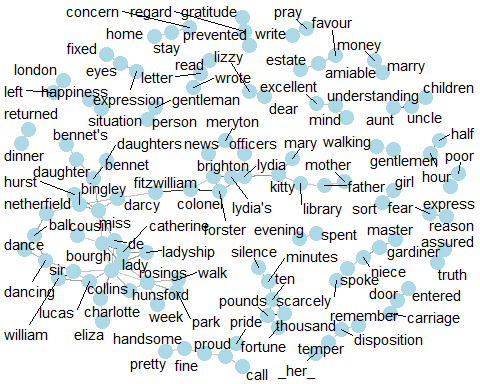
word\_cors %>%  
 filter(item1 == "pounds")

## # A tibble: 393 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 pounds thousand 0.701   
## 2 pounds ten 0.231   
## 3 pounds fortune 0.164   
## 4 pounds settled 0.149   
## 5 pounds wickham's 0.142   
## 6 pounds children 0.129   
## 7 pounds mother's 0.119   
## 8 pounds believed 0.0932  
## 9 pounds estate 0.0890  
## 10 pounds ready 0.0860  
## # ... with 383 more rows

word\_cors %>%  
 filter(item1 %in% c("elizabeth", "pounds", "married", "pride")) %>%  
 group\_by(item1) %>%  
 top\_n(6) %>%  
 ungroup() %>%  
 mutate(item2 = reorder(item2, correlation)) %>%  
 ggplot(aes(item2, correlation)) +  
 geom\_bar(stat = "identity") +  
 facet\_wrap(~ item1, scales = "free") +  
 coord\_flip()



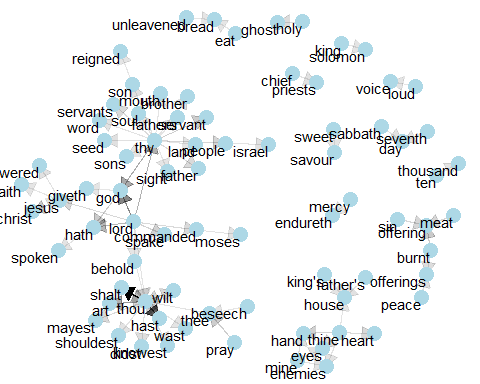
set.seed(2016)  
  
word\_cors %>%  
 filter(correlation > .15) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation), show.legend = FALSE) +  
 geom\_node\_point(color = "lightblue", size = 5) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_void()



# Make a network graph of one of the Harry Potter books or choose a different book from Gutenberg.

# the King James version is book 10 on Project Gutenberg:  
library(gutenbergr)  
kjv <- gutenberg\_download(10)

library(stringr)  
  
kjv\_bigrams <- kjv %>%  
 count\_bigrams()  
  
# filter out rare combinations, as well as digits  
kjv\_bigrams %>%  
 filter(n > 40,  
 !str\_detect(word1, "\\d"),  
 !str\_detect(word2, "\\d")) %>%  
 visualize\_bigrams()



titles <- c("Philosopher's Stone", "Chamber of Secrets", "Prisoner of Azkaban",  
 "Goblet of Fire", "Order of the Phoenix", "Half-Blood Prince",  
 "Deathly Hallows")  
  
books <- list(philosophers\_stone, chamber\_of\_secrets, prisoner\_of\_azkaban,  
 goblet\_of\_fire, order\_of\_the\_phoenix, half\_blood\_prince,  
 deathly\_hallows)  
   
hp <- tibble()  
  
for(i in seq\_along(titles)) {  
   
 clean <- tibble(chapter = seq\_along(books[[i]]),  
 text = books[[i]]) %>%  
 mutate(book = titles[i]) %>%  
 select(book, everything())  
  
 hp <- rbind(hp, clean)  
}  
  
# set factor to keep books in order of publication  
hp$book <- factor(hp$book, levels = rev(titles))  
  
hp

## # A tibble: 200 x 3  
## book chapter text   
## <fct> <int> <chr>   
## 1 Philosopher's S~ 1 "THE BOY WHO LIVED Mr. and Mrs. Dursley, of number four, Privet Drive,~  
## 2 Philosopher's S~ 2 "THE VANISHING GLASS Nearly ten years had passed since the Dursleys ha~  
## 3 Philosopher's S~ 3 "THE LETTERS FROM NO ONE The escape of the Brazilian boa constrictor e~  
## 4 Philosopher's S~ 4 "THE KEEPER OF THE KEYS BOOM. They knocked again. Dudley jerked awake.~  
## 5 Philosopher's S~ 5 "DIAGON ALLEY Harry woke early the next morning. Although he could tel~  
## 6 Philosopher's S~ 6 "THE JOURNEY FROM PLATFORM NINE AND THREE-QUARTERS Harry's last month ~  
## 7 Philosopher's S~ 7 "THE SORTING HAT The door swung open at once. A tall, black-haired wit~  
## 8 Philosopher's S~ 8 "THE POTIONS MASTER There, look.\" \"Where?\" \"Next to the tall kid~  
## 9 Philosopher's S~ 9 "THE MIDNIGHT DUEL Harry had never believed he would meet a boy he hat~  
## 10 Philosopher's S~ 10 "HALLOWEEN Malfoy couldn't believe his eyes when he saw that Harry and~  
## # ... with 190 more rows

hp\_bigrams <- hp %>%  
 count\_bigrams()  
  
# filter out rare combinations, as well as digits  
hp\_bigrams %>%  
 filter(n > 40,  
 !str\_detect(word1, "\\d"),  
 !str\_detect(word2, "\\d")) %>%  
 visualize\_bigrams()

