Some Simple SPOOKY Data Analysis

Huijun Cui

January 31, 2018

I. Prerequisite 1. Setup the libraries

```
packages.used <- c("ggplot2", "dplyr", "tibble", "tidyr",  "stringr", "tidytext", "topicmodels",
 "wordcloud", "ggridges","igraph","tweenr","ggraph","scales")

# check packages that need to be installed.
packages.needed <- setdiff(packages.used, intersect(installed.packages()[,1], packages.used))

# install additional packages
if(length(packages.needed) > 0) {
  install.packages(packages.needed, dependencies = TRUE, repos = 'http://cran.us.r-project.org')
}

library(ggplot2)
library(dplyr)
library(tibble)
library(tidyr)
library(stringr)
library(tidytext)
#install.packages("topicmodels")
library(topicmodels)
library(wordcloud)
library(ggridges)
#install.packages("igraph")
library(igraph)
#install.packages("tweenr")
#library(tweenr)
#install.packages("ggraph")
library(ggraph)
library(scales)

source("../libs/multiplot.R")
```

2. Read in the data

```
spk <- read.csv('../data/spooky.csv', as.is = TRUE)
```

3. An overview of the data structure and content

```
head(spk)
```

```
##        id
## 1 id26305
## 2 id17569
## 3 id11008
## 4 id27763
## 5 id12958
## 6 id22965
##
                                                                                          text
## 1
                                                  This process, however, afforded me no means of
ascertaining the dimensions of my dungeon; as I might make its circuit, and return to the point
whence I set out, without being aware of the fact; so perfectly uniform seemed the wall.
## 2
                  It never once occurred to me that the fumbling might be a mere mistake.
## 3
                                                                              In his left han
d was a gold snuff box, from which, as he capered down the hill, cutting all manner of fantastic
steps, he took snuff incessantly with an air of the greatest possible self satisfaction.
## 4
                                                                        How lovely is spring
As we looked from Windsor Terrace on the sixteen fertile counties spread beneath, speckled by ha
ppy cottages and wealthier towns, all looked as in former years, heart cheering and fair.
## 5
           Finding nothing else, not even gold, the Superintendent abandoned his attempts; but a
perplexed look occasionally steals over his countenance as he sits thinking at his desk.
## 6 A youth passed in solitude, my best years spent under your gentle and feminine fosterage, h
as so refined the groundwork of my character that I cannot overcome an intense distaste to the u
sual brutality exercised on board ship: I have never believed it to be necessary, and when I hea
rd of a mariner equally noted for his kindliness of heart and the respect and obedience paid to
him by his crew, I felt myself peculiarly fortunate in being able to secure his services.
##   author
## 1    EAP
## 2    HPL
## 3    EAP
## 4    MWS
## 5    HPL
## 6    MWS
```

```
summary(spk)
```

```
##       id                 text               author
##  Length:19579       Length:19579       Length:19579
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

Each row of the dataset contains a unique ID, a single sentence text excerpt, and an abbreviated author name. `HPL` is Lovecraft, `MWS` is Shelly, and `EAP` is Poe.

```
sum(is.na(spk))
```

```
## [1] 0
```

```
spk$author <- as.factor(spk$author)
```

Thus, there are no missing values. And the author name is transformed to be a factor variable.

II. Data Cleaning

The `unnest_tokens()` function to drop all punctuation and transform all words into lower case. At least for now, the punctuation isn't really important to our analysis – we want to study the words. In addition, `tidytext` contains a dictionary of stop words, like "and" or "next", that we will get rid of for our analysis, the idea being that the non-common words (…maybe the SPOOKY words) that the authors use will be more interesting.

```
#library(janeaustenr)
library(dplyr)
library(stringr)

spk_byauthor <- spk %>%
  group_by(author) %>%
  mutate(linenumber = row_number()) %>%
  ungroup()
spkline <- spk %>%
  mutate(line = row_number())

spk_wrd <- unnest_tokens(spk_byauthor, word, text)
spk_wrdn <- unnest_tokens(spkline, word, text) %>%
  group_by(line)%>%
  mutate(wordorder = row_number())%>%
  ungroup
spk_wrd <- spk_wrd %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

III. Data analysis

1. Stop word analysis

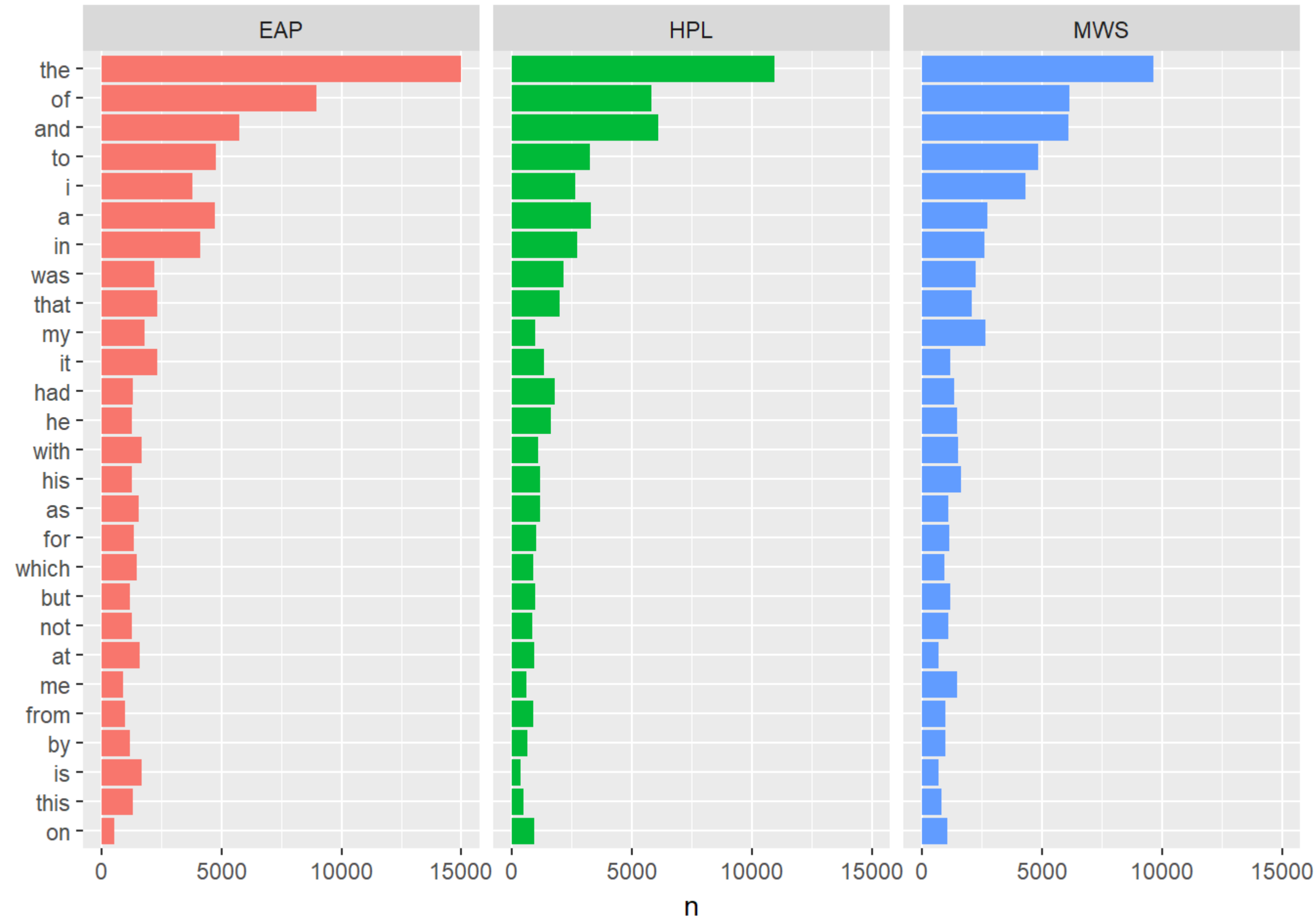
```r
#library(janeaustenr)
library(dplyr)
library(stringr)

spk_byauthor <- spk %>%
  group_by(author) %>%
  mutate(linenumber = row_number()) %>%
  ungroup()

spk_wrd <- unnest_tokens(spk_byauthor, word, text)
spk_stp <- unnest_tokens(spk_byauthor, word, text) %>% semi_join(stop_words)
```

```
## Joining, by = "word"
```

```r
author_words <- count(group_by(spk_stp, word, author))

all_words    <- rename(count(group_by(spk_stp, word)), all = n)
author_words <- left_join(author_words, all_words, by = "word")
author_words <- arrange(author_words, desc(all))
author_words <- ungroup(head(author_words, 81))
author_words
```

```
## # A tibble: 81 x 4
##    word  author     n   all
##    <chr> <fct>  <int> <int>
##  1 the   EAP    14993 35585
##  2 the   HPL    10933 35585
##  3 the   MWS     9659 35585
##  4 of    EAP     8972 20955
##  5 of    HPL     5846 20955
##  6 of    MWS     6137 20955
##  7 and   EAP     5735 17956
##  8 and   HPL     6098 17956
##  9 and   MWS     6123 17956
## 10 to    EAP     4765 12842
## # ... with 71 more rows
```

```r
#png("../figs/stopword_frequency.png")
ggplot(author_words) +
  geom_col(aes(reorder(word, all, FUN = min), n, fill = author)) +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ author) +
  theme(legend.position = "none")
```




```
summary(spk_wrd)
```

```
##        id               author          linenumber        word          
##  Length:522984      EAP:200965   Min.   :   1   Length:522984     
##  Class :character   HPL:156319   1st Qu.:1630   Class :character  
##  Mode  :character   MWS:165700   Median :3239   Mode  :character  
##                                  Mean   :3322                     
##                                  3rd Qu.:4886                     
##                                  Max.   :7900                     
```

```
summary(spk_byauthor)
```

```
##        id                text            author       linenumber  
##  Length:19579       Length:19579       EAP:7900   Min.   :   1  
##  Class :character   Class :character   HPL:5635   1st Qu.:1632  
##  Mode  :character   Mode  :character   MWS:6044   Median :3264  
##                                                   Mean   :3338  
##                                                   3rd Qu.:4895  
##                                                   Max.   :7900  
```

The picture shows the habits of the three writers when they use stop words, and the second table provides the precise values of frequency of stop words. In the table, n is the stop word frequency of each author, while all refers to the frequency of all three authors. From these materials, we can find generally authors have similar tendency in stop word using, but typically EAP like to utilize stop words more than other two authors(especially in the example of word ‘the’). However, the summary also shows that there are more sentences and words from EAP in the dataset, therefore the distribution may also result from this quantitative superiority.

2. First word analysis

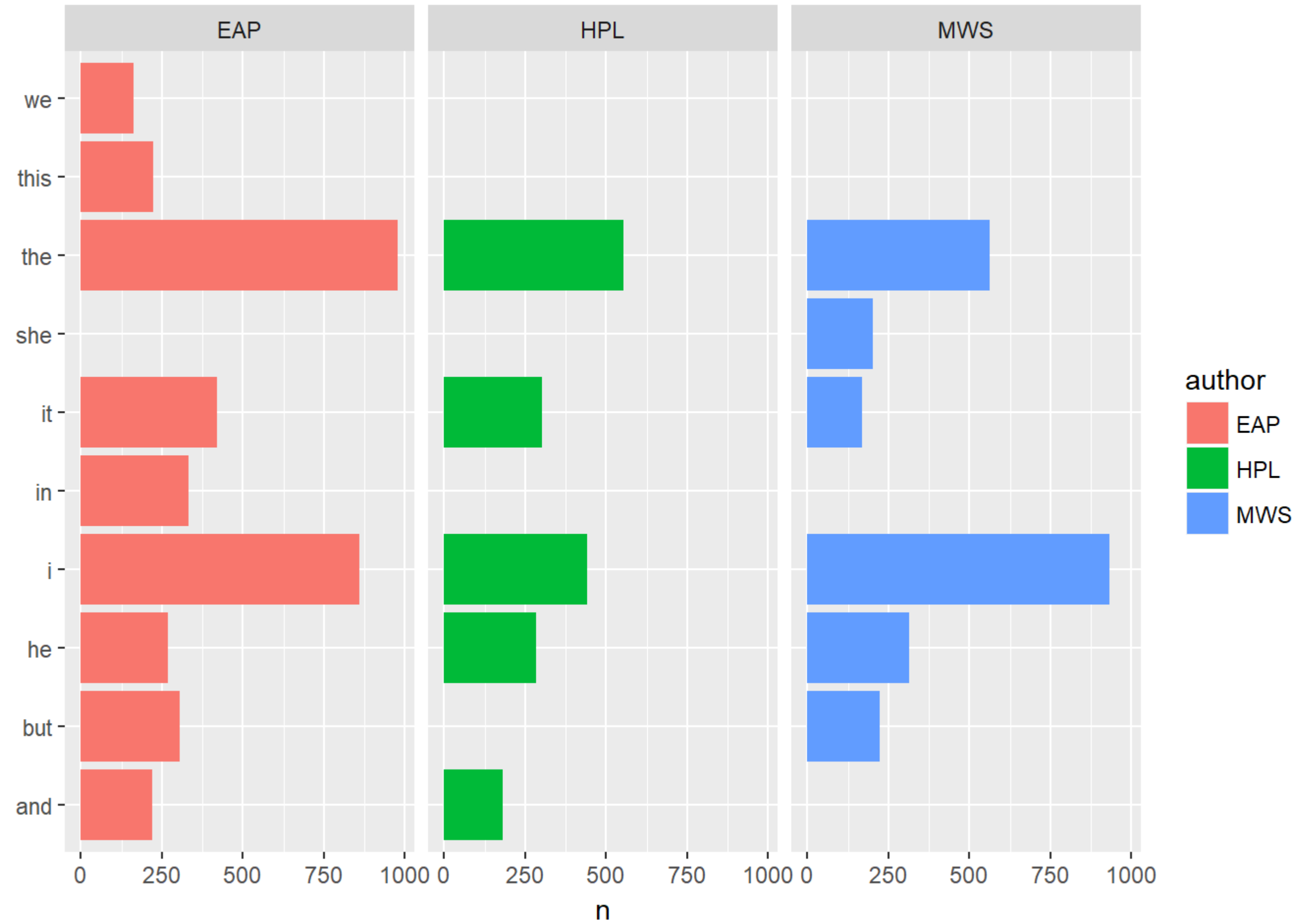
```
spkline <- spk %>%
  mutate(line = row_number())

spk_wrdn <- unnest_tokens(spkline, word, text) %>%
  group_by(line)%>%
  mutate(wordorder = row_number())%>%
  ungroup

first_wrd <- filter(spk_wrdn, wordorder == 1)
first_wrd
```

```
## # A tibble: 19,579 x 5
##    id      author  line word    wordorder
##    <chr>   <fct>  <int> <chr>       <int>
##  1 id26305 EAP        1 this            1
##  2 id17569 HPL        2 it              1
##  3 id11008 EAP        3 in              1
##  4 id27763 MWS        4 how             1
##  5 id12958 HPL        5 finding         1
##  6 id22965 MWS        6 a               1
##  7 id09674 EAP        7 the             1
##  8 id13515 EAP        8 the             1
##  9 id19322 EAP        9 i               1
## 10 id00912 MWS       10 i               1
## # ... with 19,569 more rows
```

```
#png("../figs/firstword_distribution.png")
first_wrd %>%
  group_by(author)%>%
  count(word, sort = TRUE) %>%
  head(20) %>%
  mutate(first = reorder(word, n)) %>%
  ggplot(aes(first, n, fill = author)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ author)
```

```
## Warning in mutate_impl(.data, dots): Unequal factor levels: coercing to
## character
```

```
## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector

## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector

## Warning in mutate_impl(.data, dots): binding character and factor vector,
## coercing into character vector
```



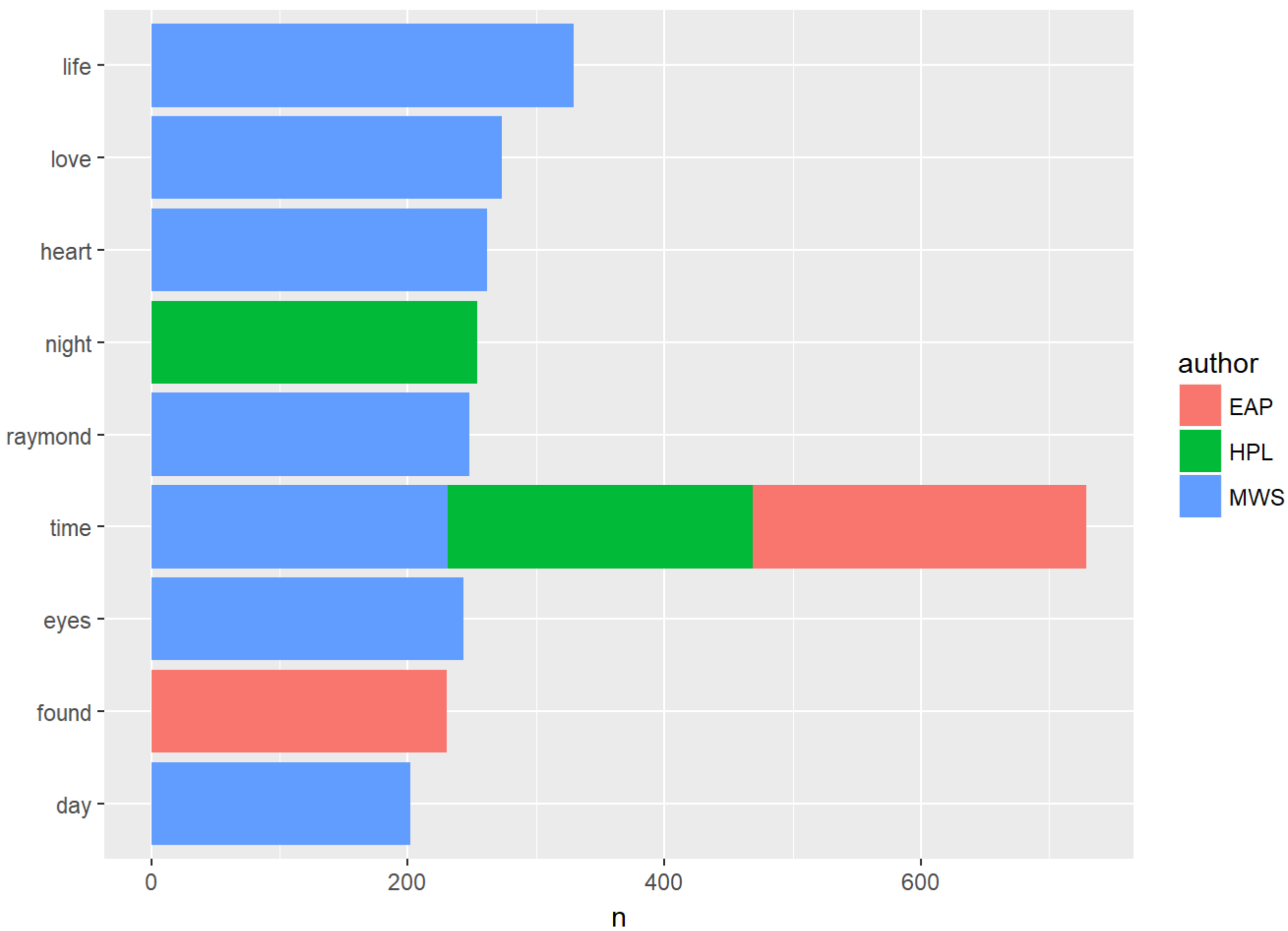
The table above shows the occupation modes of the first word of each sentence, and the table describes the first word(wih the top 20 frequencies) utilization distribution of each authors. They show that most of first words are stop words, so the word distribution also corresponds to the results in stop word analysis – the words EAP utilized as first words occupies more percentage in the most popular first word group.

3. Word frequency analysis

```
library(ggplot2)

spk_wrd <- spk_wrd %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
spk_count <- spk_wrd %>%
  group_by(author) %>%
  count(word, sort = TRUE) %>%
  ungroup
#png("../figs/topwords.png")
spk_count %>%
  filter(n > 200) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = author)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
spk_count
```

```
## # A tibble: 40,185 x 3
##    author word        n
##    <fct>  <chr>   <int>
##  1 MWS    life      329
##  2 MWS    love      273
##  3 MWS    heart     262
##  4 EAP    time      260
##  5 HPL    night     254
##  6 MWS    raymond   248
##  7 MWS    eyes      243
##  8 HPL    time      238
##  9 MWS    time      231
## 10 EAP    found     230
## # ... with 40,175 more rows
```

These gragh and table give us the description of word analysis without consideration about stop words. From both the gragh and the table, it is obvious that MWS contributes most words (no stop word) in the dataset, though for some special cases like ‘time’, ‘night’, ‘found’ and etc, EAP and HPL dominate more. From this phenomena we can induce that MWS does not like to use stop words compared with other two authors.

```
spk_eap <- filter(spk_wrd, author == "EAP")
spk_hpl <- filter(spk_wrd, author == "HPL")
spk_mws <- filter(spk_wrd, author == "MWS")

spk_eap %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 14,868 x 2
##    word       n
##    <chr>  <int>
##  1 time     260
##  2 found    230
##  3 length   178
##  4 day      174
##  5 eyes     168
##  6 head     164
##  7 night    143
##  8 left     140
##  9 matter   139
## 10 de       133
## # ... with 14,858 more rows
```

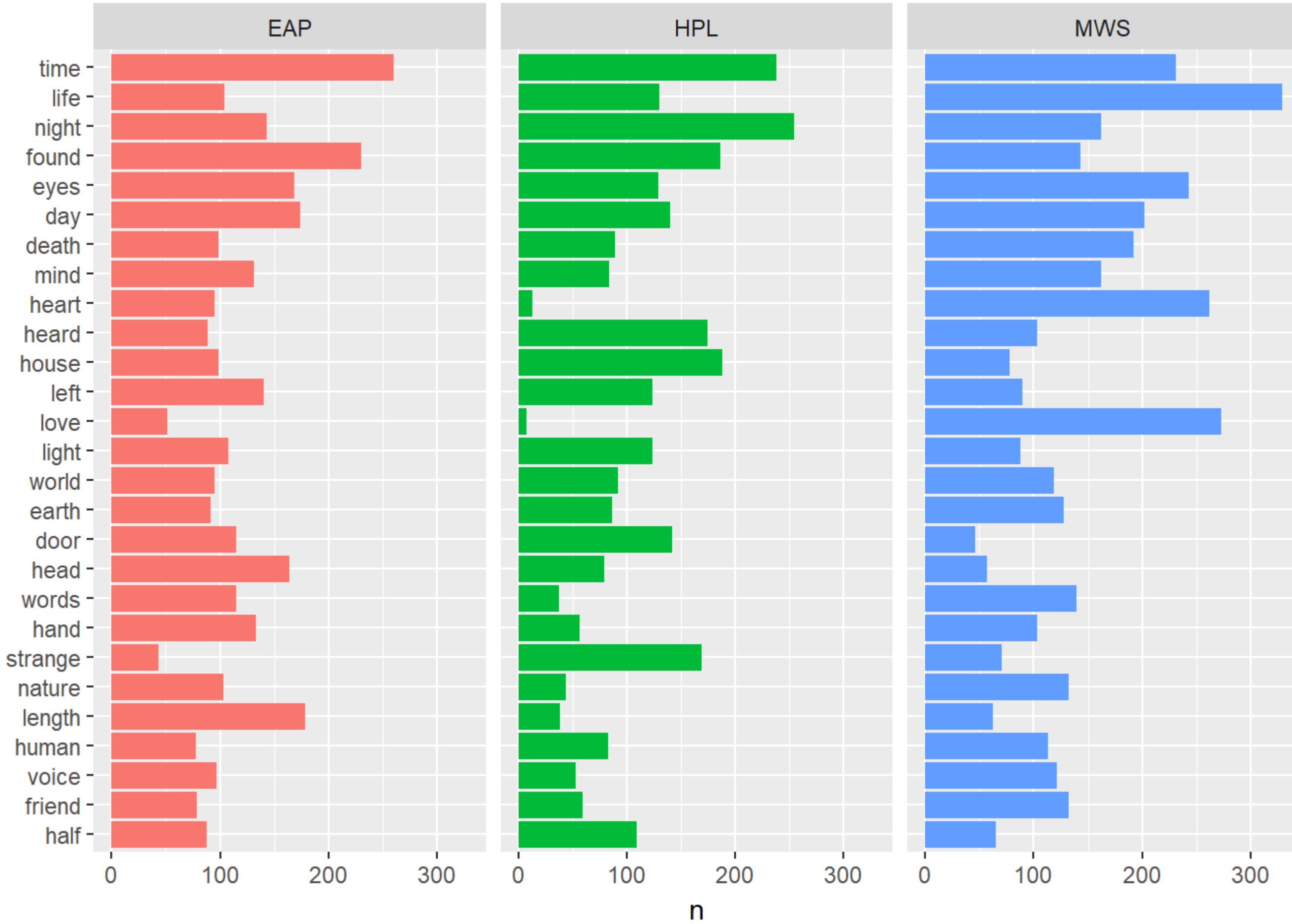
```
spk_hpl %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 14,202 x 2
##    word        n
##    <chr>   <int>
##  1 night     254
##  2 time      238
##  3 house     188
##  4 found     186
##  5 heard     174
##  6 strange   169
##  7 street    146
##  8 told      143
##  9 door      142
## 10 day       140
## # ... with 14,192 more rows
```

```
spk_mws %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 11,115 x 2
##    word        n
##    <chr>   <int>
##  1 life      329
##  2 love      273
##  3 heart     262
##  4 raymond   248
##  5 eyes      243
##  6 time      231
##  7 day       202
##  8 death     192
##  9 father    173
## 10 mind      162
## # ... with 11,105 more rows
```

```
# Counts number of times each author used each word.
author_words <- count(group_by(spk_wrd, word, author))

# Counts number of times each word was used.
all_words    <- rename(count(group_by(spk_wrd, word)), all = n)

author_words <- left_join(author_words, all_words, by = "word")
author_words <- arrange(author_words, desc(all))
author_words <- ungroup(head(author_words, 81))
#png("../figs/topword_distribution.png")
ggplot(author_words) +
  geom_col(aes(reorder(word, all, FUN = min), n, fill = author)) +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ author) +
  theme(legend.position = "none")
```



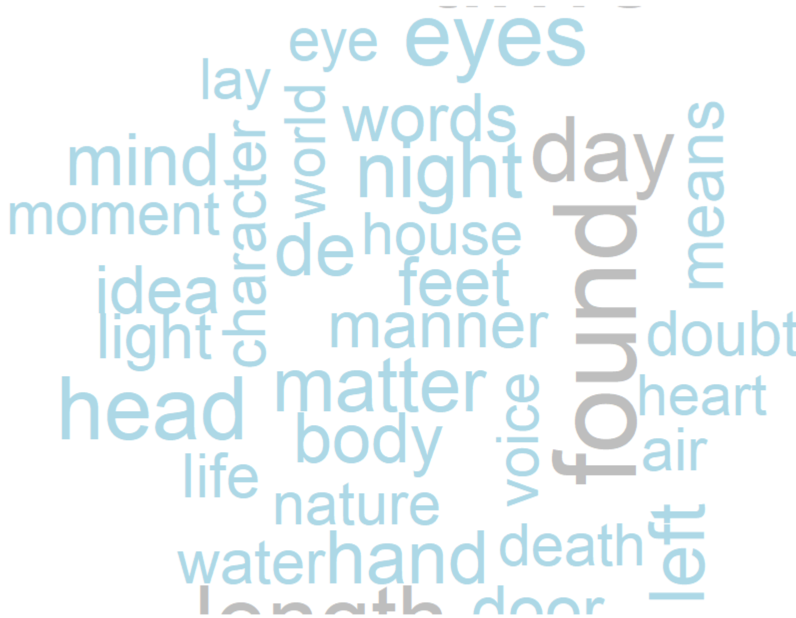
This part is about the detailed word profiles of the authors. We can see the wording preference of difference writers, like MWS tends to talk about life, while HPL and EAP prefer to discuss about time, MWS loves to mention love, while the other two authors are not. The first table is the word and its frequency list of EAP, the second table is for HPL, and the third one is for MWS.

```
# Words is a list of words, and freqs their frequencies
spk_wrd <- spk_wrd %>% anti_join(stop_words)
```

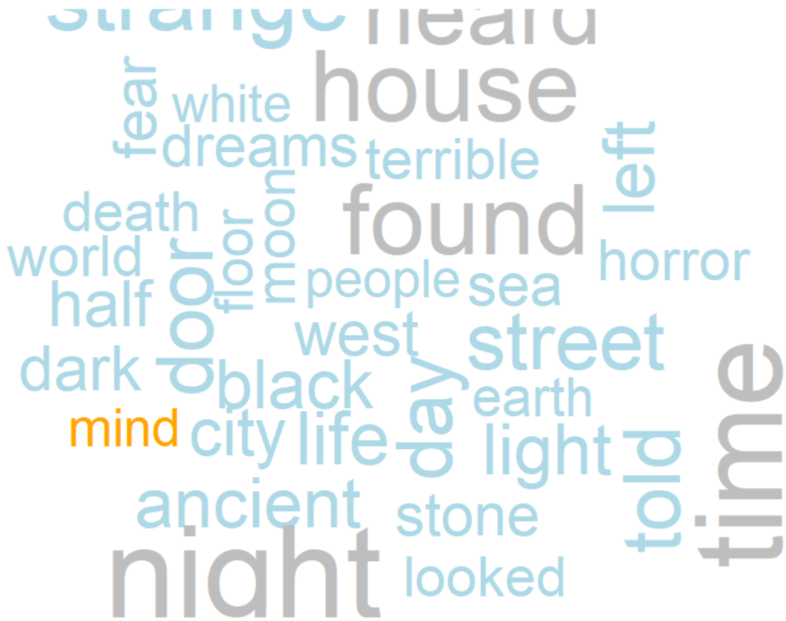
```
## Joining, by = "word"
```

```
wordstotal <- count(group_by(spk_wrd, word))$word
freqstotal <- count(group_by(spk_wrd, word))$n

wordseap <- count(group_by(spk_eap, word))$word
freqseap <- count(group_by(spk_eap, word))$n

wordshpl <- count(group_by(spk_hpl, word))$word
freqshpl <- count(group_by(spk_hpl, word))$n

wordsmws <- count(group_by(spk_mws, word))$word
freqsmws <- count(group_by(spk_mws, word))$n

#png("../figs/Wordcloud_all.png")
wordcloud(wordstotal, freqstotal, max.words = 35, color = c("orange", "lightblue","grey"))
```


```
#png("../figs/Wordcloud_eap.png")
wordcloud(wordseap, freqseap, max.words = 35, color = c("orange", "lightblue","grey"))
```




```
#png("../figs/Wordcloud_hpl.png")
wordcloud(wordshpl, freqshpl, max.words = 35, color = c("orange", "lightblue","grey"))
```




```
#png("../figs/Wordcloud_mws.png")
wordcloud(wordsmws, freqsmws, max.words = 35, color = c("orange", "lightblue","grey"))
```


```
#dev.off()
```

Here are the word cloud graphs, in whih the 35 most common words in the entire datset and personal dataset of each writer are plotted. It is very intuitionistic that “time”, “life”, and “night” all appear frequently.

4. Correlation analysis from data of different authoers

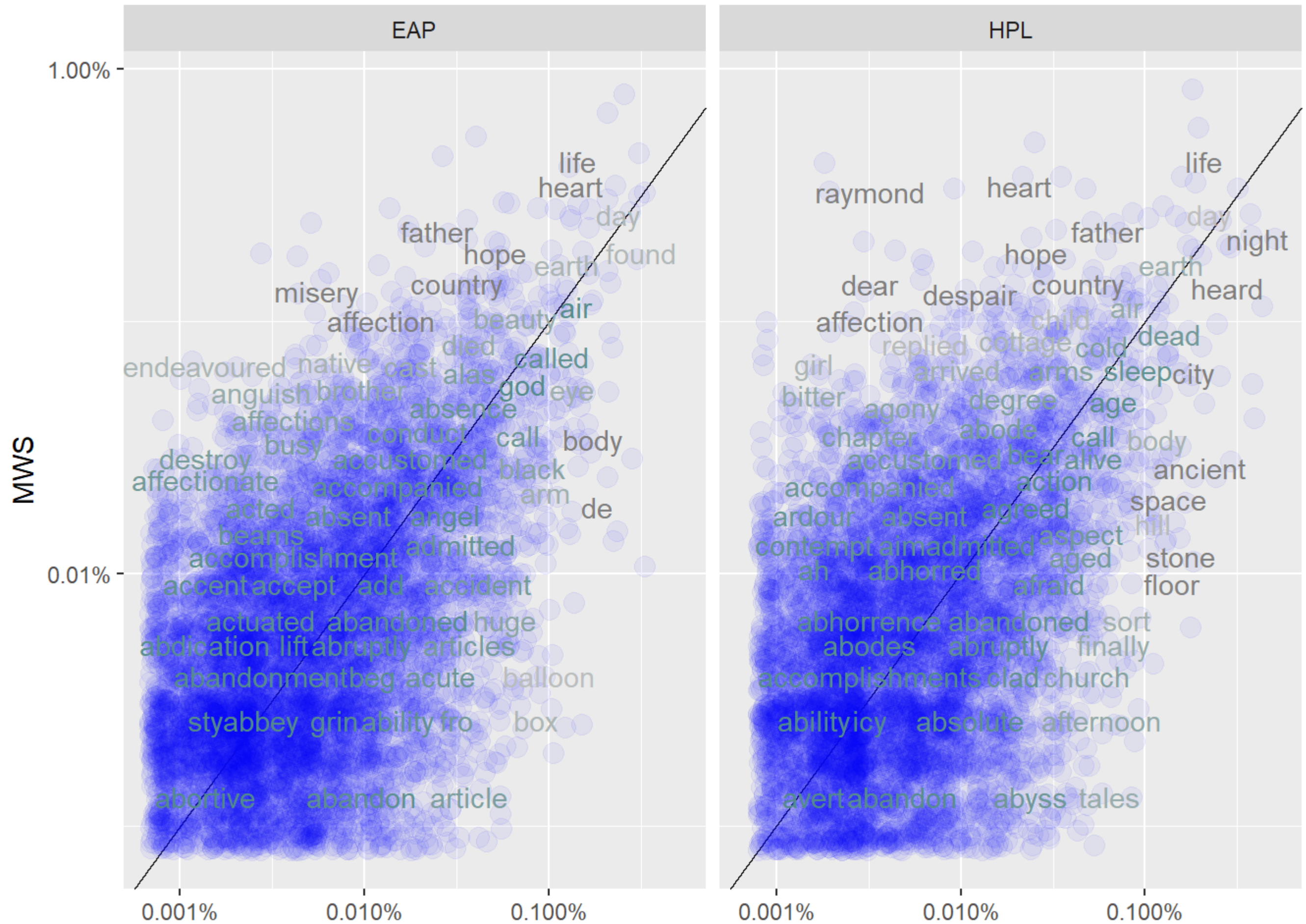
```
frequency <- spk_wrd %>%
  #extract words from possible italics
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion,'EAP':'HPL')

frequency
```

```
## # A tibble: 49,858 x 4
##    word               MWS author proportion
##    <chr>            <dbl> <chr>       <dbl>
##  1 a           NA         EAP     0.0000412
##  2 aaem        NA         EAP     0.0000137
##  3 ab          NA         EAP     0.0000137
##  4 aback       NA         EAP     0.0000274
##  5 abaft        0.0000160 EAP    NA
##  6 abandon      0.0000160 EAP     0.0000960
##  7 abandoned    0.0000800 EAP     0.000151
##  8 abandoning  NA         EAP     0.0000274
##  9 abandonment  0.0000480 EAP     0.0000274
## 10 abaout      NA         EAP    NA
## # ... with 49,848 more rows
```

```
library(scales)

# expect a warning about rows with missing values being removed
#png("../figs/frequencycorrelationMWS.png")
ggplot(frequency, aes(x = proportion, y = MWS, color = abs(MWS - proportion))) +
  geom_abline(color = "gray10", lty = 1) +
  geom_jitter(alpha = 0.05, color = 12, size = 3.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  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) +
  theme(legend.position="none") +
  labs(y = "MWS", x = NULL)
```

```
## Warning: Removed 36887 rows containing missing values (geom_point).
```

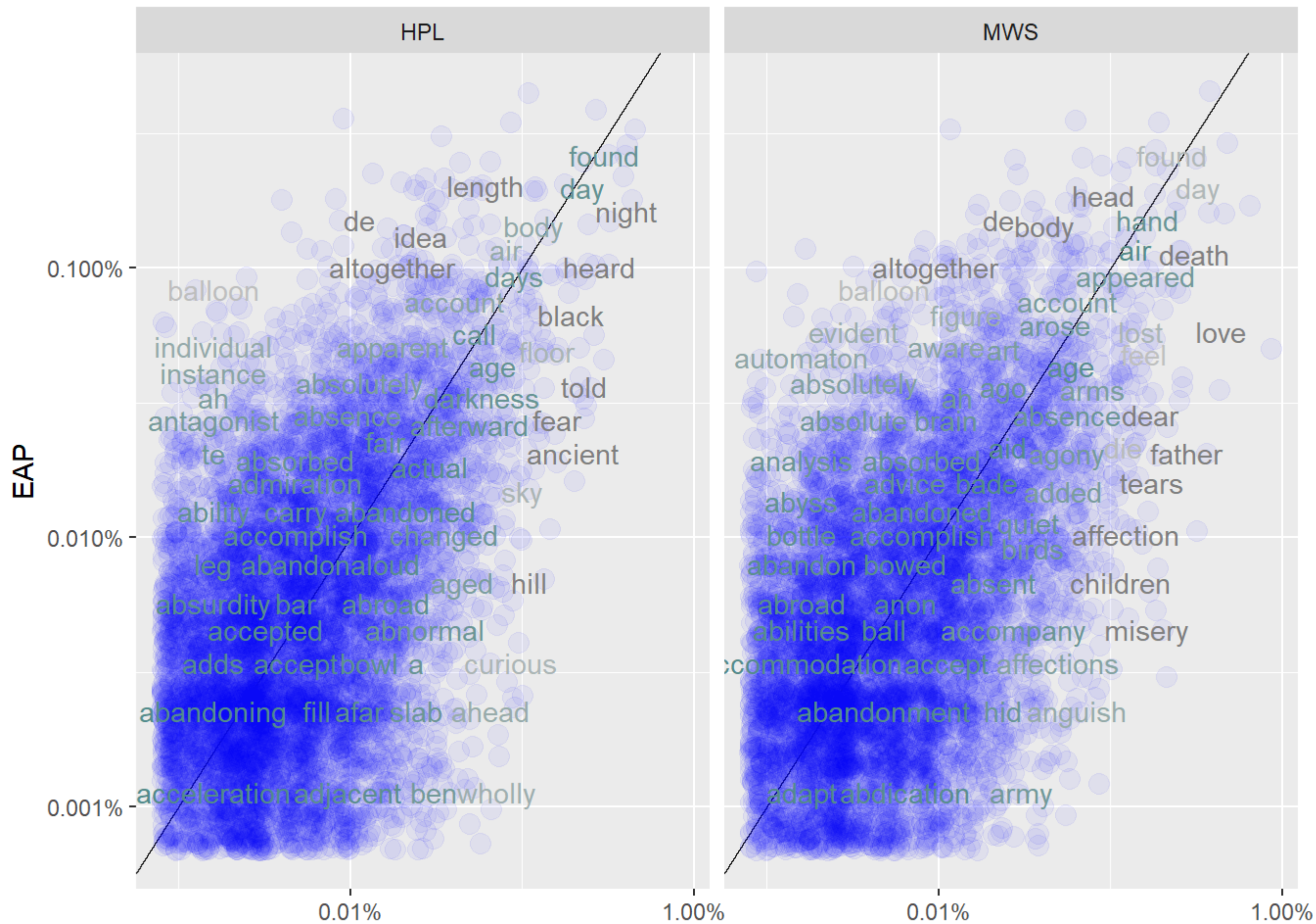
```
## Warning: Removed 36887 rows containing missing values (geom_text).
```




```
frequency2 <- spk_wrd %>%
  #extract words from possible italics
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion,'MWS':'HPL')

library(scales)
# expect a warning about rows with missing values being removed
#png("../figs/frequencycorrelationeap.png")
ggplot(frequency2, aes(x = proportion, y = EAP, color = abs(EAP - proportion))) +
  geom_abline(color = "gray10", lty = 1) +
  geom_jitter(alpha = 0.05, color = 12, size = 3.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  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) +
  theme(legend.position="none") +
  labs(y = "EAP", x = NULL)
```

```
## Warning: Removed 35907 rows containing missing values (geom_point).
```

```
## Warning: Removed 35908 rows containing missing values (geom_text).
```



Here are the word frequency comparison graphs between EAP&HPL, EAP&MWS, MWS&HPL, in which the coordinate values represent the proportion of words in the text groups of different authors. From the graphs, the word points close to the abline if they have similar proportion value in the text groups of different authors. For example, in the graph EAP&HPL and EAP&MWS, we can pick up the word ‘together’, which is above the abline on both of the graphs. This means ‘together’ has a relatively higher proportion in EAP’s works than it in HPL&MWS’s works.

```
cor.test(data = frequency[frequency$author == "EAP",],
         ~ proportion + MWS)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and MWS
## t = 65.709, df = 6800, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6084326 0.6375106
## sample estimates:
##       cor
## 0.6231869
```

```
cor.test(data = frequency[frequency$author == "HPL",],
         ~ proportion + MWS)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and MWS
## t = 53.903, df = 6167, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5487050 0.5826377
## sample estimates:
##      cor
## 0.565911
```

```
cor.test(data = frequency2[frequency2$author == "HPL",],
         ~ proportion + EAP)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and EAP
## t = 68.623, df = 7147, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6160482 0.6440021
## sample estimates:
##       cor
## 0.6302294
```

The first correlation test is between EAP and MWS, the second one is between HPL and MWS, the last one is between HPL and EAP. These results show the correlation relationship between HPL&MWS is lower than that between HPL&EAP and that between MWS&EAP.

5. TF-IDF

TF-IAF shows the relative frequency a certain author uses a word compared with that all the authors use the word, and this can be regarded as a more detailed edition of the last part.

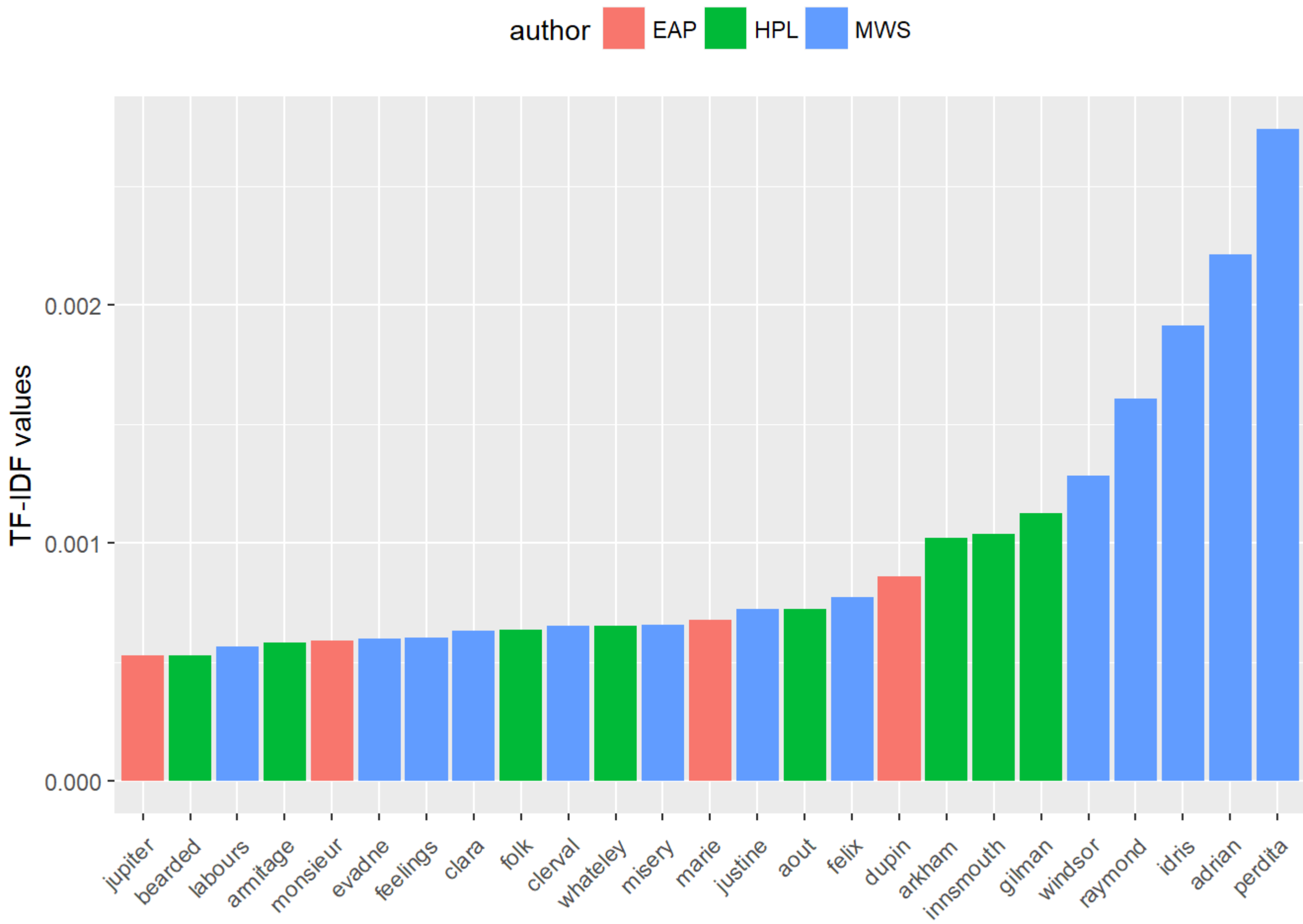
```
frequency <- count(spk_wrd, author, word)
tf_idf    <- bind_tf_idf(frequency, word, author, n)
head(tf_idf)
```

```
## # A tibble: 6 x 6
##   author word        n        tf   idf    tf_idf
##   <fct>  <chr>   <int>     <dbl> <dbl>     <dbl>
## 1 EAP    ã          19 0.000261  0.405 0.000106
## 2 EAP    a.m         3 0.0000412 0.405 0.0000167
## 3 EAP    aaem        1 0.0000137 1.10  0.0000151
## 4 EAP    ab          1 0.0000137 1.10  0.0000151
## 5 EAP    aback       2 0.0000274 1.10  0.0000301
## 6 EAP    abandon     7 0.0000960 0     0
```

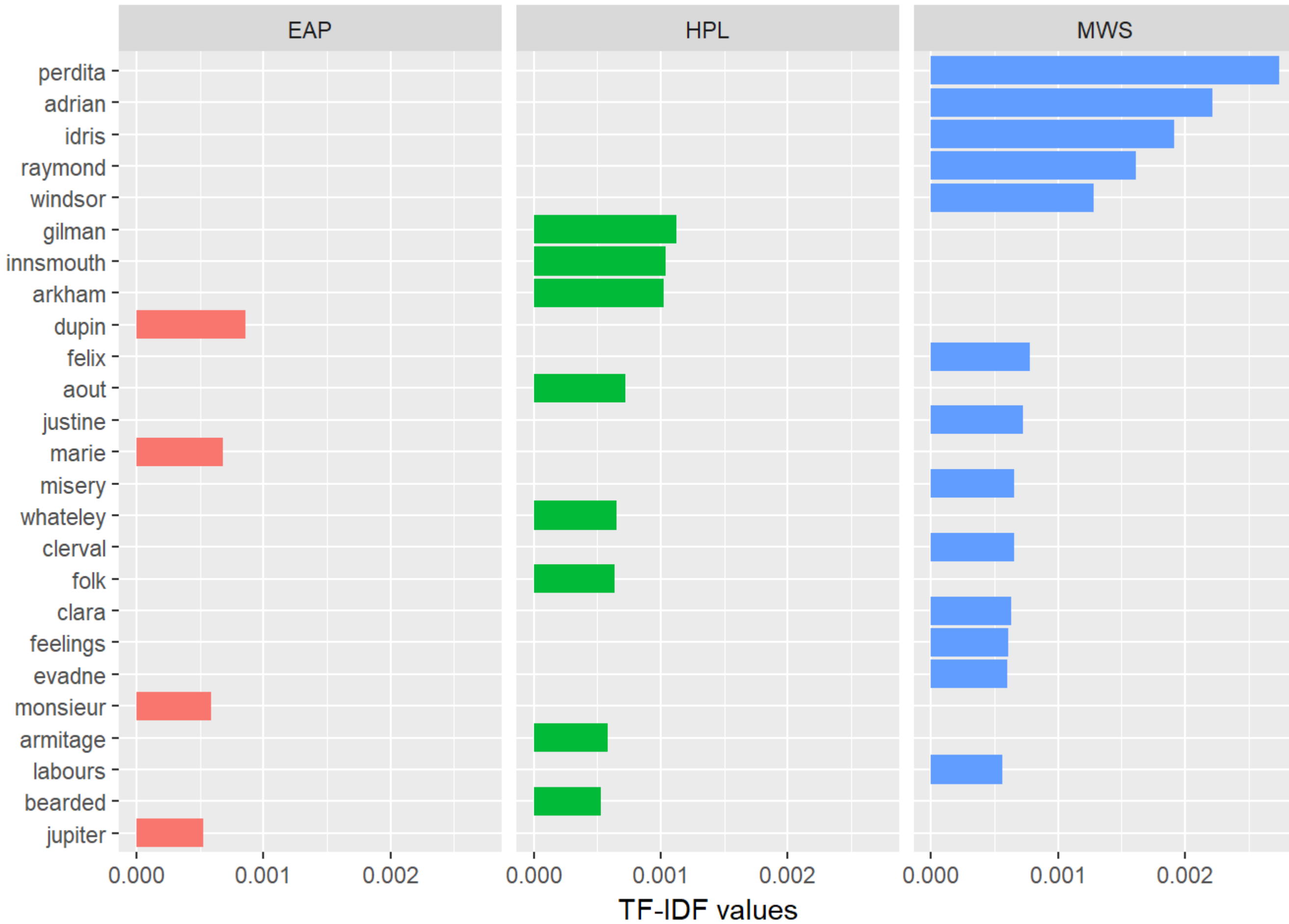
```
tail(tf_idf)
```

```
## # A tibble: 6 x 6
##   author word         n        tf   idf     tf_idf
##   <fct>  <chr>    <int>     <dbl> <dbl>      <dbl>
## 1 MWS    youth's      1 0.0000160 0.405 0.00000649
## 2 MWS    youthful    10 0.000160  0     0
## 3 MWS    youths       2 0.0000320 0.405 0.0000130
## 4 MWS    zaimi        2 0.0000320 1.10  0.0000352
## 5 MWS    zeal         7 0.000112  0     0
## 6 MWS    zest         3 0.0000480 0     0
```

```
tf_idf    <- arrange(tf_idf, desc(tf_idf))
tf_idf    <- mutate(tf_idf, word = factor(word, levels = rev(unique(word))))

tf_idf_25 <- top_n(tf_idf, 25, tf_idf)
#png("../figs/tf_idf_25.png")
ggplot(tf_idf_25) +
  geom_col(aes(word, tf_idf, fill = author)) +
  labs(x = NULL, y = "TF-IDF values") +
  theme(legend.position = "top", axis.text.x  = element_text(angle=45, hjust=1, vjust=0.9))
```




```r
#png("../figs/tf_idf_25sep.png")
ggplot(tf_idf_25) +
  geom_col(aes(word, tf_idf, fill = author)) +
  labs(x = NULL, y = "TF-IDF values") +
  coord_flip() +
  facet_wrap(~ author) +
  theme(legend.position = "none")
```



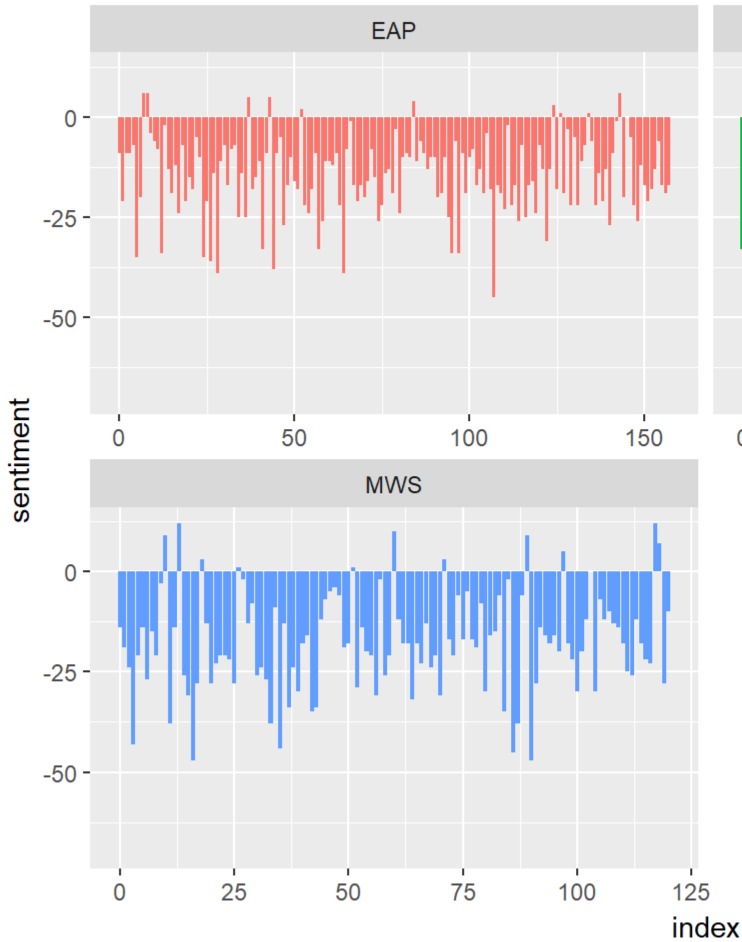
The first table is the head part of TF-IDF distribution table. and the second one is the tail part of the TF-IDF list. From the distribution histogram, the typical words are usually names or nouns, which are related to different topics and contents different author like to discuss. They can be regarded as signs to recognize who the author is of a certain text.

6. Sentiment Analysis

```
#library(janeaustenr)
library(dplyr)
library(stringr)
library(tidyr)

samplesentiment <- spk_wrd %>%
  inner_join(get_sentiments("bing")) %>%
  count(author, index = linenumber %/% 50, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
library(ggplot2)
#png("../figs/sentimentbing.png")

ggplot(samplesentiment, aes(index, sentiment, fill = author)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~author, ncol = 2, scales = "free_x")
```



Here

I use Bing et al. as the lexicon to analyse the emotional bias. I use the value of positive - negative as the representative of sentiment, and put every 50 sentences as a group to make analysis. From the graphs above, we can conclude most of the time all of the authors discussed about negative things, especially HPL.

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

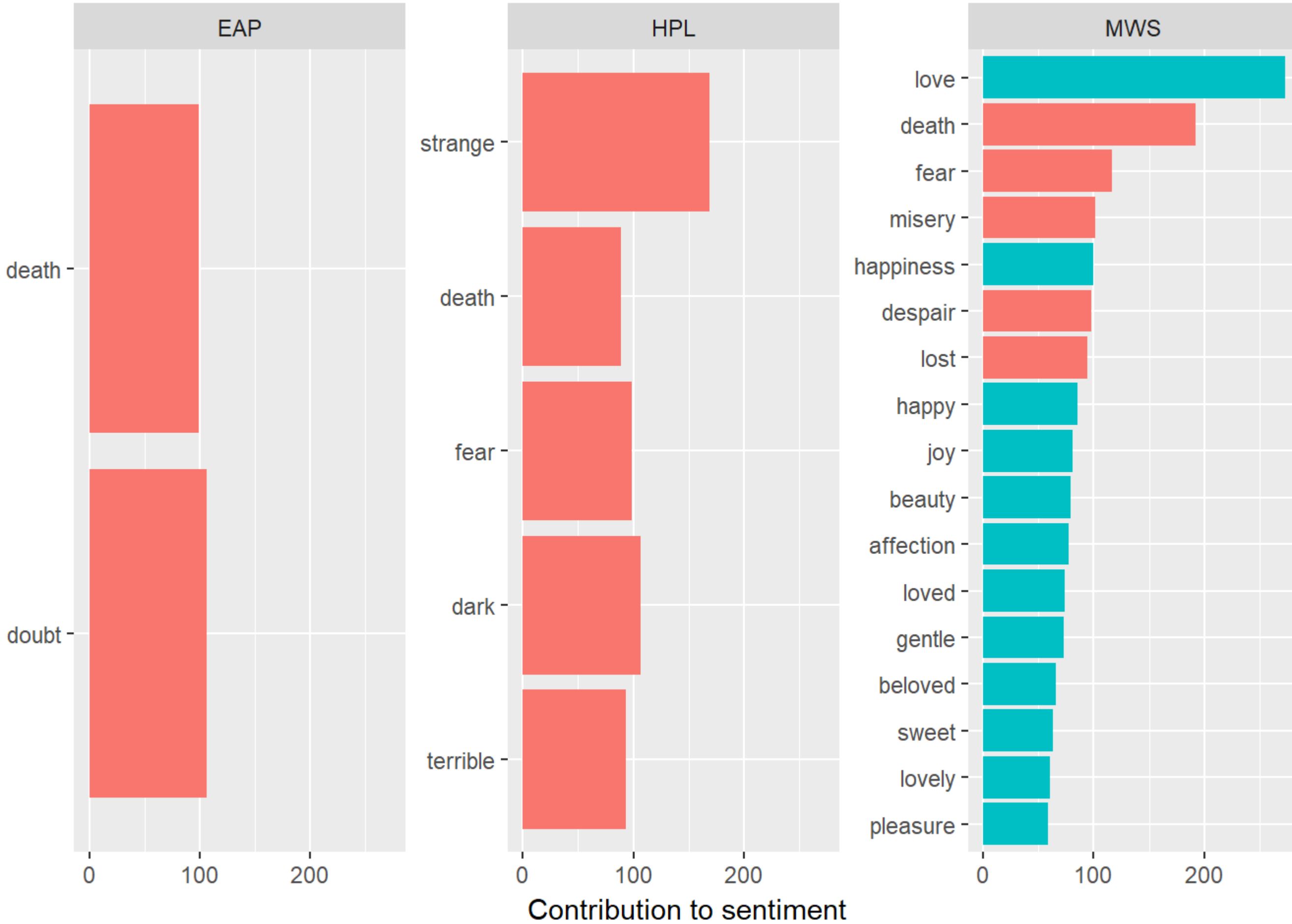
```
## Joining, by = "word"
```

```
bing_word_counts
```

```
## # A tibble: 6,799 x 4
##    word      author sentiment     n
##    <chr>     <fct>  <chr>     <int>
##  1 love      MWS    positive    273
##  2 death     MWS    negative    192
##  3 strange   HPL    negative    169
##  4 fear      MWS    negative    117
##  5 dark      HPL    negative    107
##  6 doubt     EAP    negative    106
##  7 misery    MWS    negative    101
##  8 happiness MWS    positive    100
##  9 death     EAP    negative     99
## 10 fear      HPL    negative     99
## # ... with 6,789 more rows
```

```
#png("../figs/sentimenttopword.png")
bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(12) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~author, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip()
```

```
## Selecting by n
```



Here

are the distribution of emotional words. The table shows the words, their sentiment categories, and the quantities, and their author. The graph is more straightforward, and it shows the words with sentiment of top 12 frequencies in the whole dataset. We can observe that MWS use most sentimental words, and in contrary, HPL and EAP have a more calm and cold writing style. Besides, MWS prefer to use more warm and positive words, while the other two authors like to use negative words.

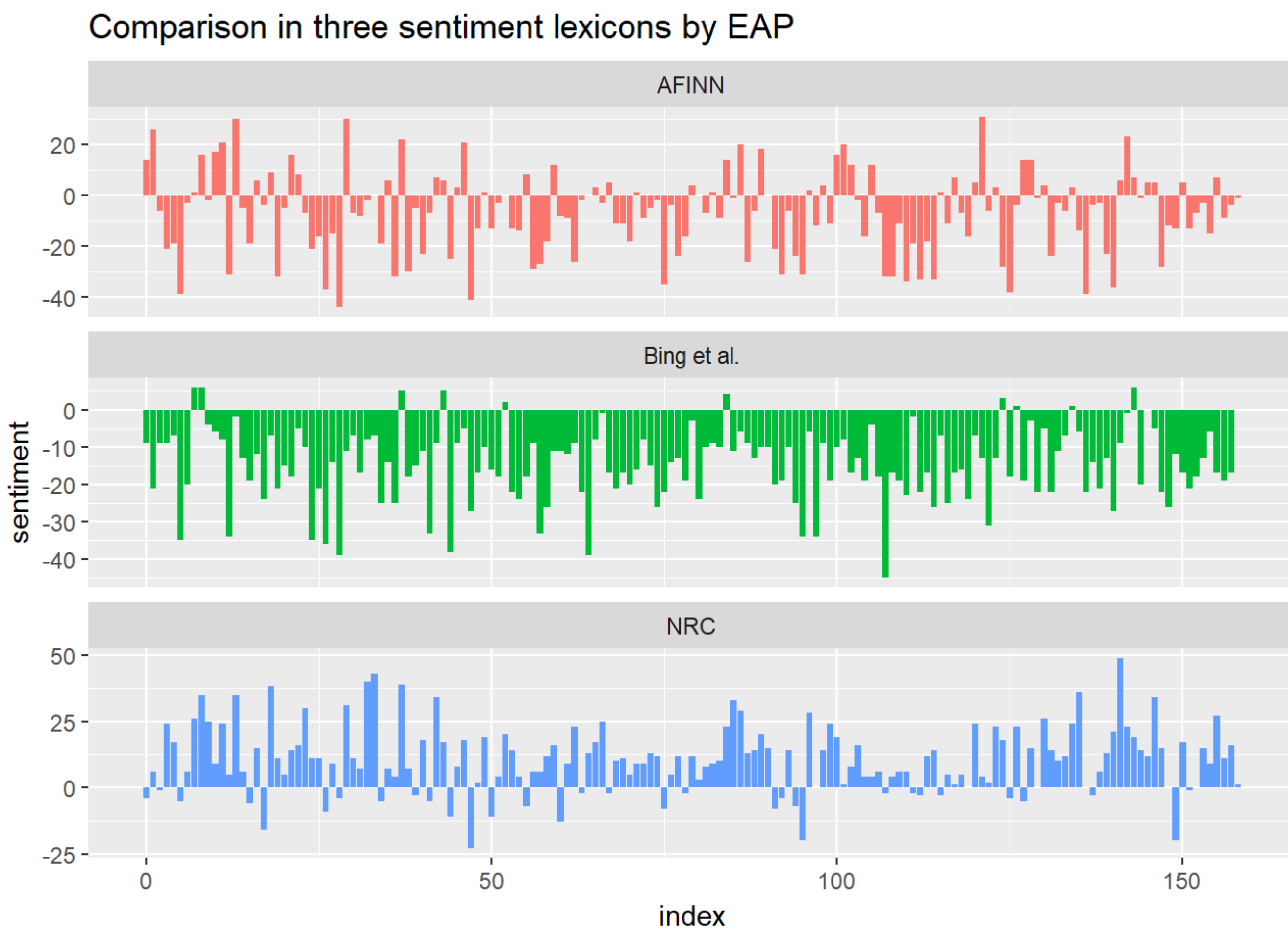
```
afinneap <- spk_eap %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 50) %>%
  summarise(sentiment = sum(score)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
bing_and_nrceap <- bind_rows(spk_eap %>%
                            inner_join(get_sentiments("bing")) %>%
                            mutate(method = "Bing et al."),
                          spk_eap %>%
                            inner_join(get_sentiments("nrc") %>%
                                         filter(sentiment %in% c("positive",
                                                                 "negative"))) %>%
                            mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 50, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
#png("../figs/sentimentcompeap.png")
bind_rows(afinneap,
          bing_and_nrceap) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y") +
  ggtitle("Comparison in three sentiment lexicons by EAP", subtitle = NULL)
```




```
afinnhpl <- spk_hpl %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 50) %>%
  summarise(sentiment = sum(score)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
bing_and_nrchpl <- bind_rows(spk_hpl %>%
                            inner_join(get_sentiments("bing")) %>%
                            mutate(method = "Bing et al."),
                          spk_hpl %>%
                            inner_join(get_sentiments("nrc") %>%
                                         filter(sentiment %in% c("positive",
                                                                 "negative"))) %>%
                            mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 50, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
#png("../figs/sentimentcomphpl.png")
bind_rows(afinnhpl,
          bing_and_nrchpl) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y") +
  ggtitle("Comparison in three sentiment lexicons by HPL", subtitle = NULL)
```

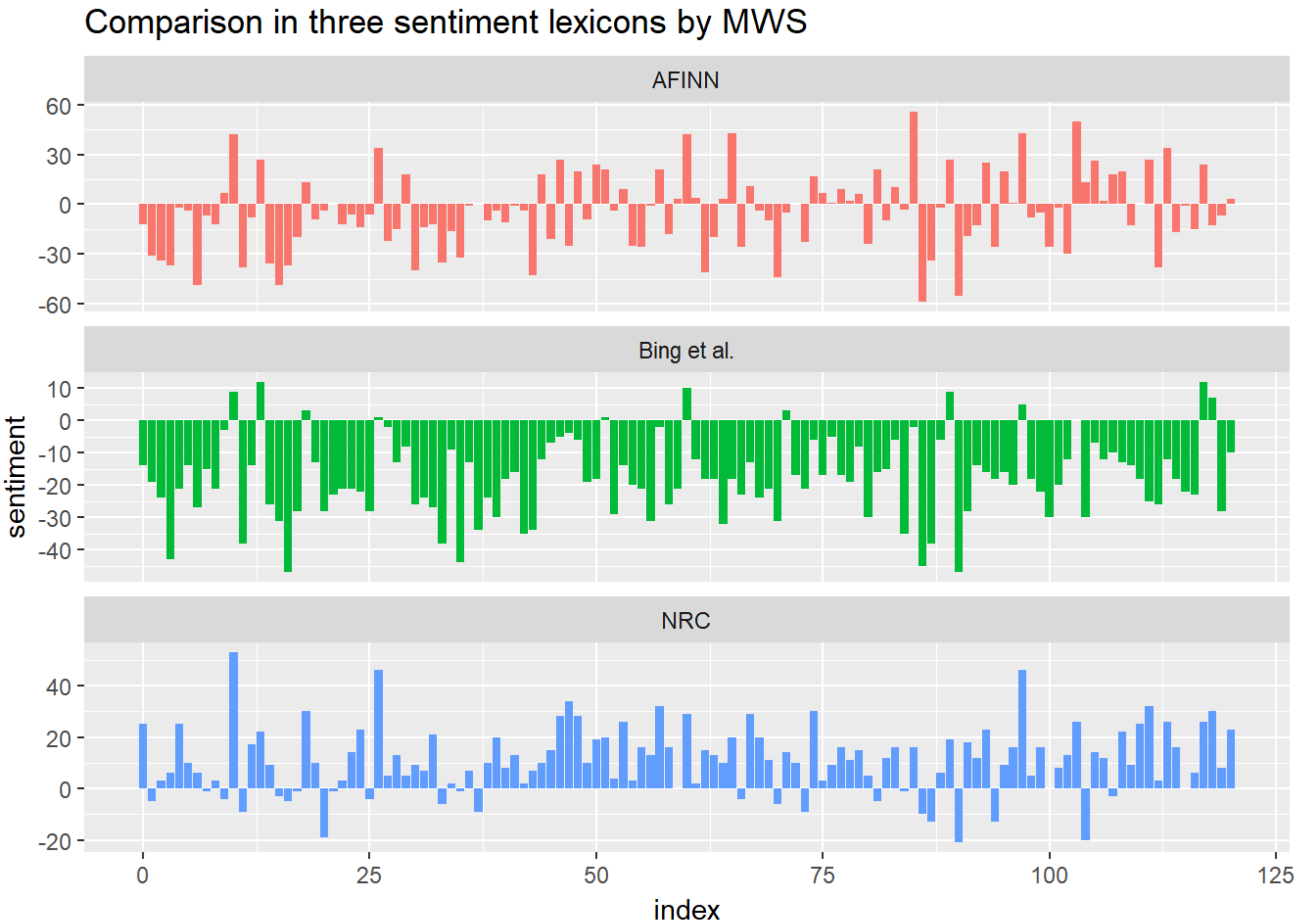
```
afinnmws <- spk_mws %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 50) %>%
  summarise(sentiment = sum(score)) %>%
  mutate(method = "AFINN")
```

```
## Joining, by = "word"
```

```
#png("../figs/sentimentcompmws.png")
bing_and_nrcmws <- bind_rows(spk_mws %>%
                            inner_join(get_sentiments("bing")) %>%
                            mutate(method = "Bing et al."),
                          spk_mws %>%
                            inner_join(get_sentiments("nrc") %>%
                                         filter(sentiment %in% c("positive",
                                                                  "negative"))) %>%
                            mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 50, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```
bind_rows(afinnmws,
          bing_and_nrcmws) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y") +
  ggtitle("Comparison in three sentiment lexicons by MWS", subtitle = NULL)
```




```
get_sentiments("nrc") %>%
     filter(sentiment %in% c("positive",
                              "negative")) %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   3324
## 2 positive   2312
```

```
get_sentiments("bing") %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4782
## 2 positive   2006
```

Here are the comparison of three diferent sentimental evaluation methods. So basically we can see they shows similar tendency in their discription, but with different sentimental scores. Usually Bing procides the lowest scores, and NRC gives us the highest scores. The last two tables show Bing has more negative words than NRC, and this

may be the reason of the discrepancy.
Also fro the graphs, HPL shows stable negative psychological state, MWS and EAP are similar, and EAP’s emotion is more turbulent than other’s.

7. Relationships between 2-grams

```
library(dplyr)
library(tidytext)
#library(janeaustenr)

eap_sentence <- filter(spk_byauthor, author == "EAP")
hpl_sentence <- filter(spk_byauthor, author == "HPL")
mws_sentence <- filter(spk_byauthor, author == "MWS")

eap_bigrams <- eap_sentence %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
hpl_bigrams <- hpl_sentence %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
mws_bigrams <- mws_sentence %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
spk_bigrams <- spk_byauthor %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)

eap_bigrams %>%
  count(bigram, sort = TRUE)
```

```
## # A tibble: 95,814 x 2
##    bigram       n
##    <chr>    <int>
##  1 of the    2877
##  2 in the    1237
##  3 to the     823
##  4 of a       530
##  5 to be      431
##  6 and the    428
##  7 it was     419
##  8 from the   403
##  9 upon the   399
## 10 it is      362
## # ... with 95,804 more rows
```

```
hpl_bigrams %>%
  count(bigram, sort = TRUE)
```

```
## # A tibble: 85,367 x 2
##    bigram       n
##    <chr>    <int>
##  1 of the    1487
##  2 in the     901
##  3 and the    503
##  4 to the     490
##  5 on the     428
##  6 from the   350
##  7 it was     348
##  8 i had      287
##  9 at the     277
## 10 of a       277
## # ... with 85,357 more rows
```

```
mws_bigrams %>%
  count(bigram, sort = TRUE)
```

```
## # A tibble: 82,010 x 2
##    bigram       n
##    <chr>    <int>
##  1 of the    1217
##  2 in the     605
##  3 to the     534
##  4 and the    412
##  5 of my      359
##  6 on the     356
##  7 i was      330
##  8 that i     296
##  9 from the   283
## 10 i had      273
## # ... with 82,000 more rows
```

```
spk_bigrams %>%
  count(bigram, sort = TRUE)
```

```
## # A tibble: 221,753 x 2
##    bigram       n
##    <chr>    <int>
##  1 of the    5581
##  2 in the    2743
##  3 to the    1847
##  4 and the   1343
##  5 it was    1037
##  6 from the  1036
##  7 on the    1011
##  8 of a       986
##  9 i had      861
## 10 of my      812
## # ... with 221,743 more rows
```

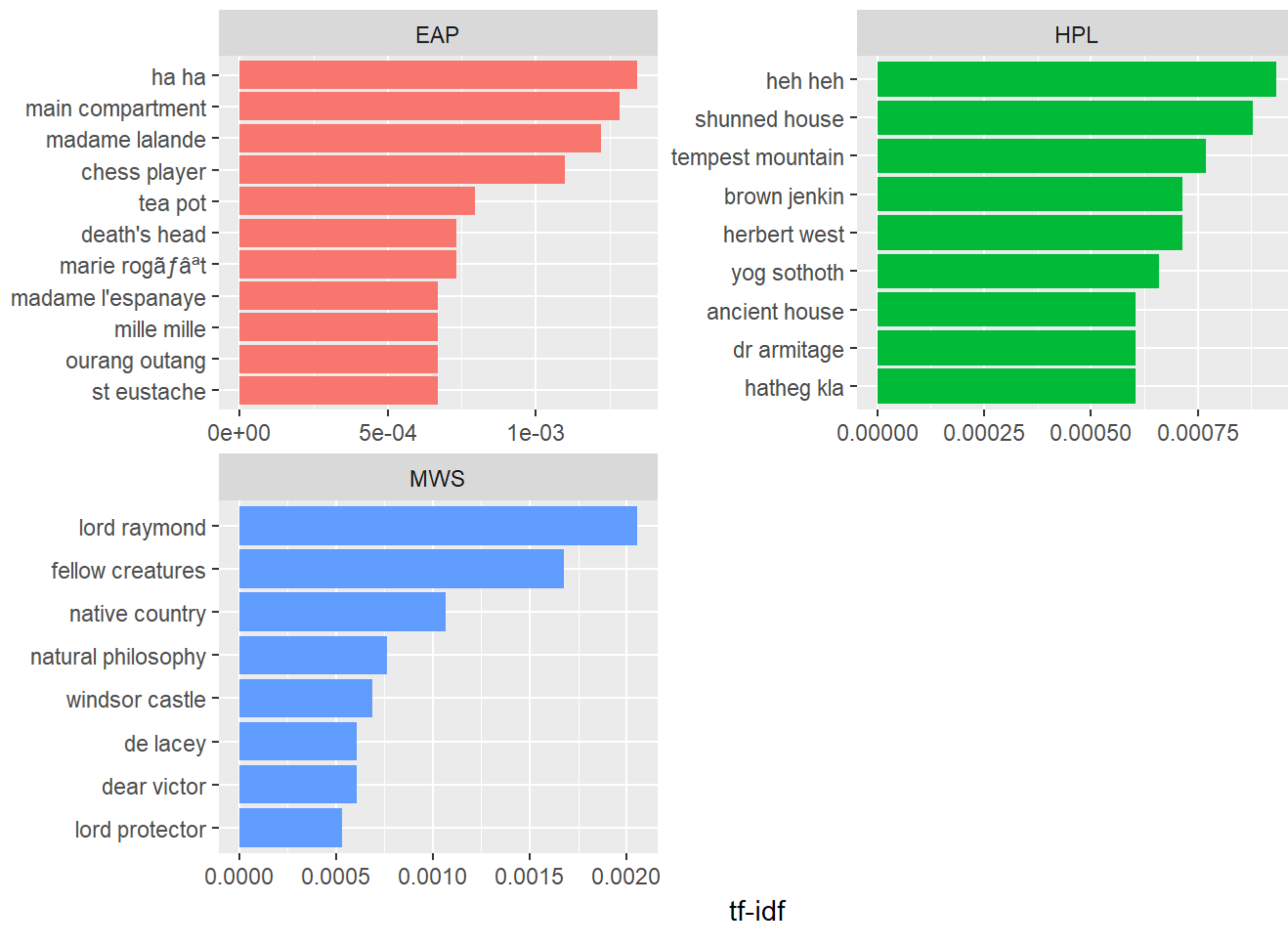
```
bigrams_separated <- spk_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

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)

bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")

bigram_counts_au <- bigrams_filtered %>%
  group_by(author) %>%
  count(word1, word2, sort = TRUE) %>%
  ungroup
bigram_counts_au
```

```
## # A tibble: 47,540 x 4
##    author word1   word2           n
##    <fct>  <chr>   <chr>       <int>
##  1 MWS    lord    raymond        27
##  2 EAP    ha      ha             22
##  3 MWS    fellow  creatures      22
##  4 EAP    main    compartment    21
##  5 EAP    madame  lalande        20
##  6 EAP    chess   player         18
##  7 HPL    heh     heh            17
##  8 HPL    shunned house          16
##  9 HPL    tempest mountain       14
## 10 MWS    native  country        14
## # ... with 47,530 more rows
```

The first three tables are bigram list of EAP, HPL and MWS, which show bigrams and their frequencies. The fourth table is a summary table of the whole dataset. Form the table, it is clear that all of authors like to use stop words ‘of the’, ‘in the’, ‘to the’ and etc. The last table is the bigram distribution without consideration of stop words. It shows usually MWS uses bigram to mention people, while others tend to use more modal particles like ‘ha ha’.

```
bigram_tf_idf <- bigrams_united %>%
  count(author, bigram) %>%
  bind_tf_idf(bigram, author, n) %>%
  arrange(desc(tf_idf))
#png("../figs/bigramtf.png")
bigram_tf_idf %>%
  mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%
  group_by(author) %>%
  top_n(8) %>%
  ungroup %>%
  ggplot(aes(bigram, tf_idf, fill = author)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~author, ncol = 2, scales = "free") +
  coord_flip()
```

```
## Selecting by tf_idf
```



They are a more straightforward presentation of the last part. Each graph shows the top bigrams each author like to use, and the bigrams also leak some details of their stories.

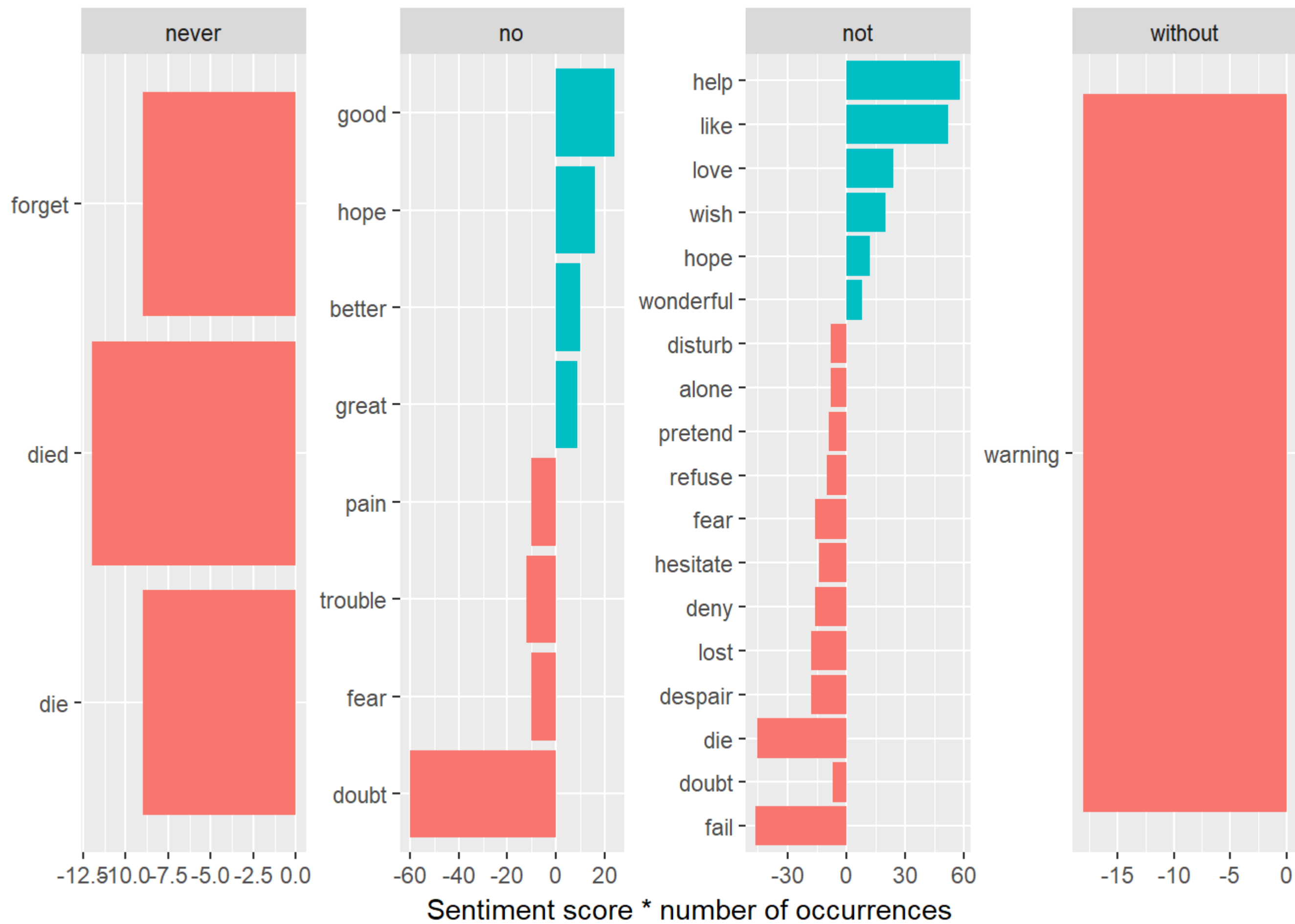
```
AFINN <- get_sentiments("afinn")
AFINN
```

```
## # 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
```

```
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) %>%
  ungroup()
negated_words
```

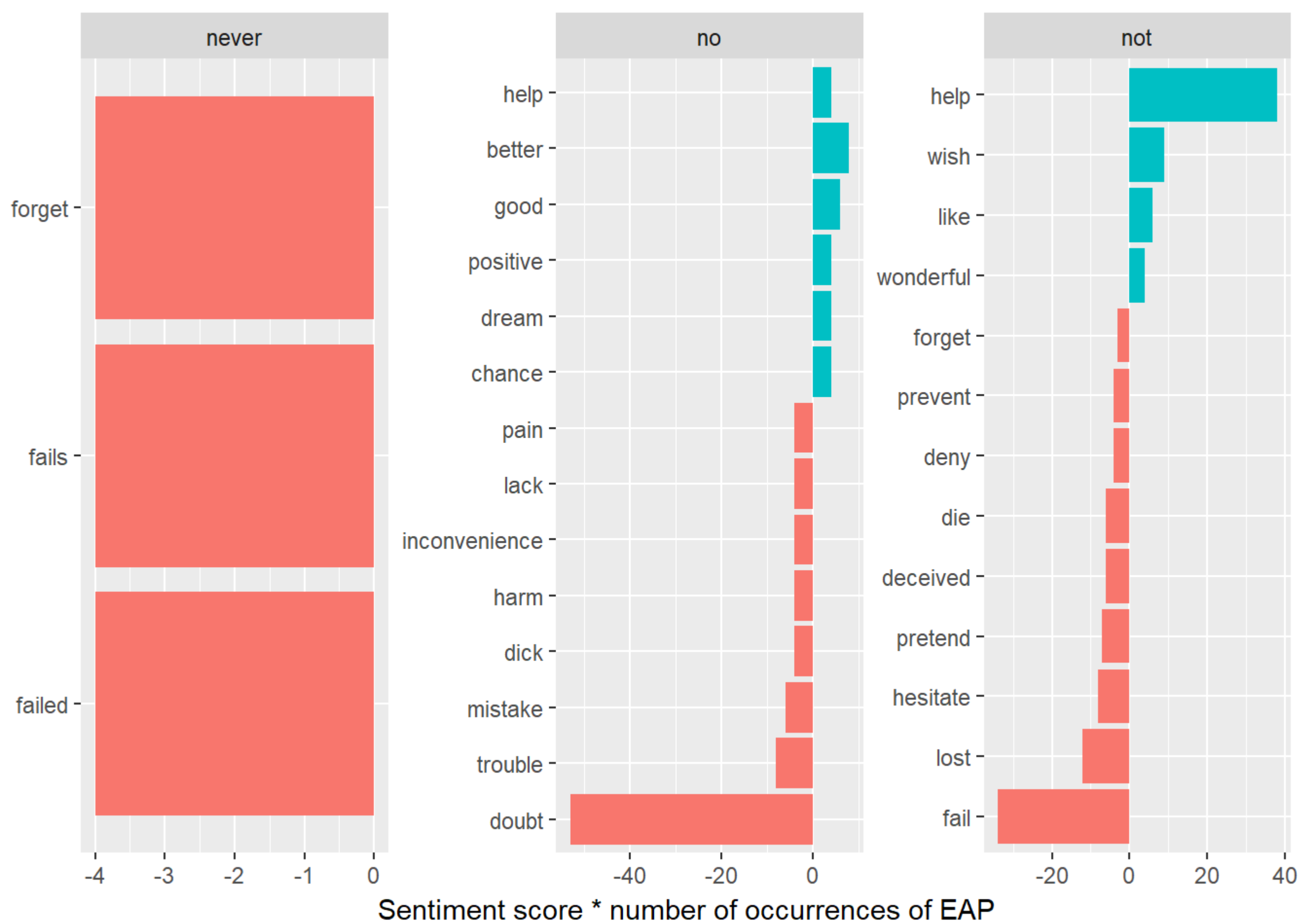
```
## # A tibble: 311 x 4
##    word1 word2   score     n
##    <chr> <chr>   <int> <int>
##  1 no    doubt      -1    60
##  2 not   help        2    29
##  3 not   like        2    26
##  4 not   fail       -2    23
##  5 not   wish        1    20
##  6 not   die        -3    15
##  7 never forget     -1     9
##  8 not   pretend    -1     9
##  9 no    good        3     8
## 10 no    hope        2     8
## # ... with 301 more rows
```

```
#png("../figs/negbigram.png")

negated_words %>%
  mutate(contribution = n * score) %>%
  group_by(word1) %>%
  arrange(desc(abs(contribution))) %>%
  head(30) %>%
  ungroup %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
  geom_col(show.legend = FALSE) +
  xlab(NULL) +
  ylab("Sentiment score * number of occurrences") +
  facet_wrap(~word1, ncol = 4, scales = "free") +
  coord_flip()
```

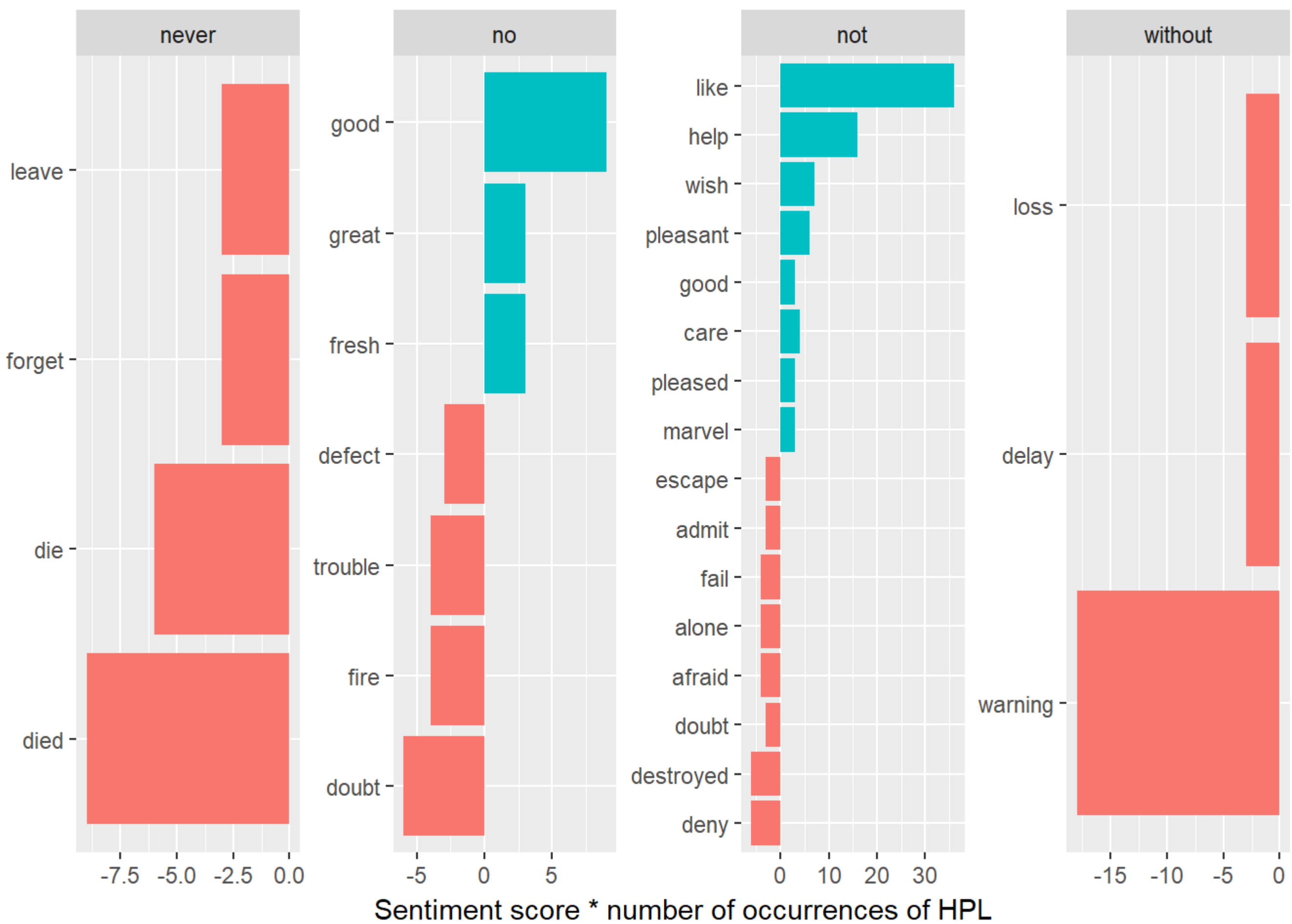


Because negation adjectives can affect the sentiment of a text, I pick up never, no, not, without to find the bigrams combined by them and sentimental words to control the influences. The lexiton utilized is afinn.
The picture above shows in the whole dataset, most of time negators are connected with a positive sentimental words. This phenomenon explains the negative sentimental style of the authors, and make us induce that may be the conclusion in the sentimental analysis is not accurate.

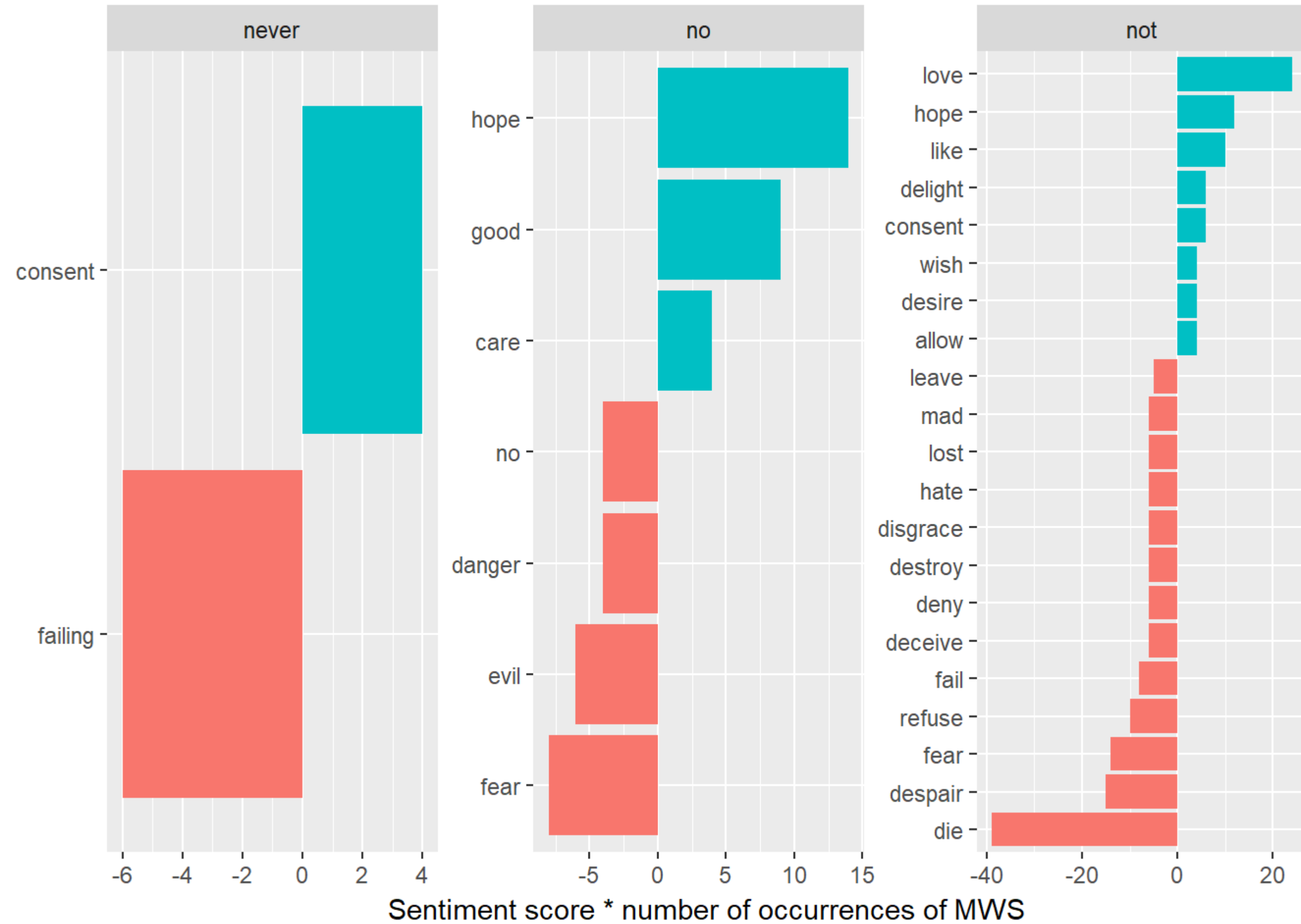
```
#EAP
eapbigrams_sep <- eap_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

negation_words <- c("not", "no", "never", "without")

negated_words <- eapbigrams_sep %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  count(word1, word2, score, sort = TRUE) %>%
  ungroup()
#png("../figs/negbigrameap.png")
negated_words %>%
  mutate(contribution = n * score) %>%
  group_by(word1) %>%
  arrange(desc(abs(contribution))) %>%
  head(30) %>%
  ungroup %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
  geom_col(show.legend = FALSE) +
  xlab(NULL) +
  ylab("Sentiment score * number of occurrences of EAP") +
  facet_wrap(~word1, ncol = 4, scales = "free") +
  coord_flip()
```



```
#HPL
hplbigrams_sep <- hpl_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

negation_words <- c("not", "no", "never", "without")

negated_words <- hplbigrams_sep %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  count(word1, word2, score, sort = TRUE) %>%
  ungroup()
#png("../figs/negbigramhpl.png")
negated_words %>%
  mutate(contribution = n * score) %>%
  group_by(word1) %>%
  arrange(desc(abs(contribution))) %>%
  head(30) %>%
  ungroup %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
  geom_col(show.legend = FALSE) +
  xlab(NULL) +
  ylab("Sentiment score * number of occurrences of HPL") +
  facet_wrap(~word1, ncol = 4, scales = "free") +
  coord_flip()
```



```
#MWS
mwsbigrams_sep <- mws_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

negation_words <- c("not", "no", "never", "without")

negated_words <- mwsbigrams_sep %>%
  filter(word1 %in% negation_words) %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  count(word1, word2, score, sort = TRUE) %>%
  ungroup()
#png("../figs/negbigrammws.png")
negated_words %>%
  mutate(contribution = n * score) %>%
  group_by(word1) %>%
  arrange(desc(abs(contribution))) %>%
  head(30) %>%
  ungroup %>%
  mutate(word2 = reorder(word2, contribution)) %>%
  ggplot(aes(word2, n * score, fill = n * score > 0)) +
  geom_col(show.legend = FALSE) +
  xlab(NULL) +
  ylab("Sentiment score * number of occurrences of MWS") +
  facet_wrap(~word1, ncol = 4, scales = "free") +
  coord_flip()
```

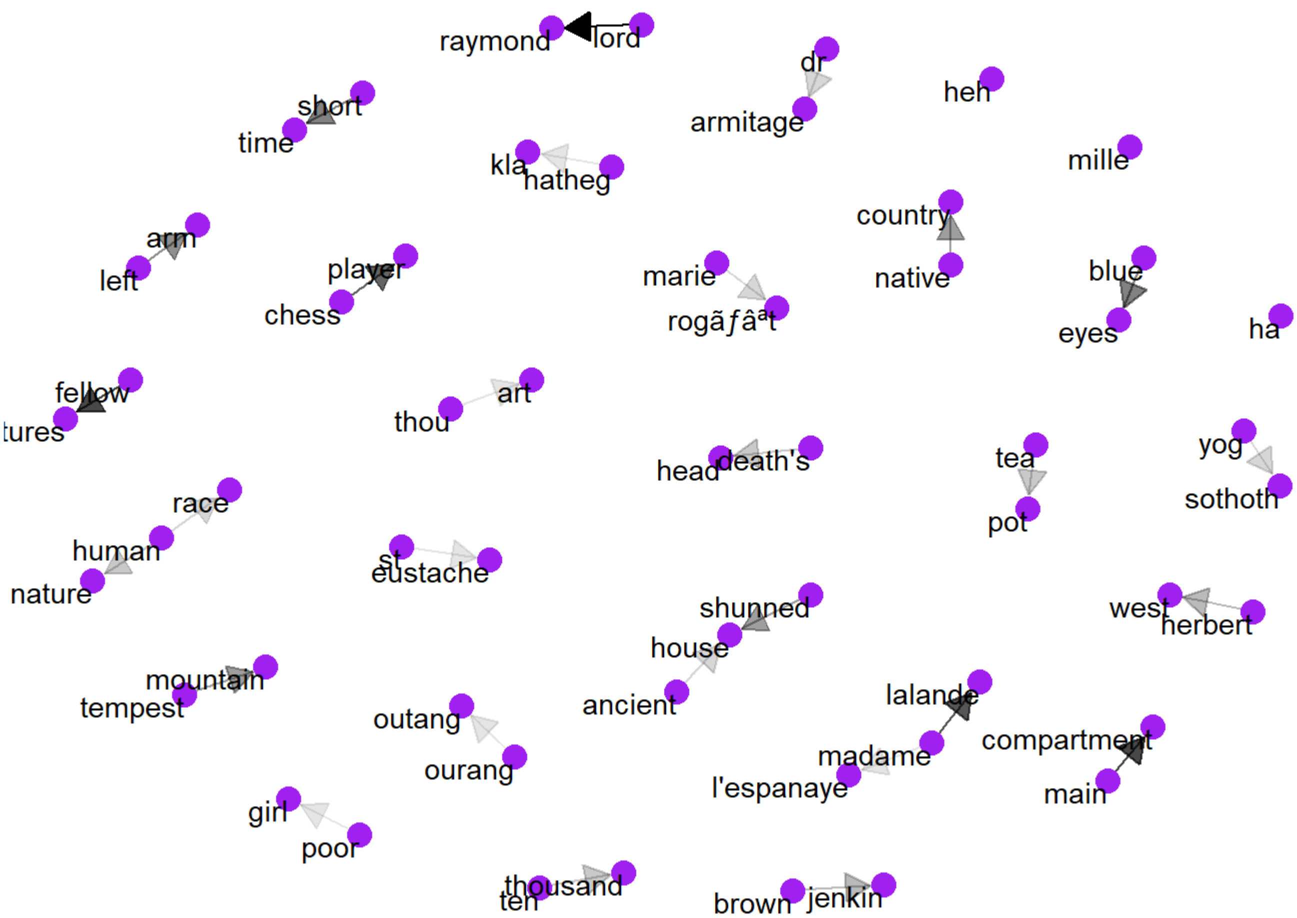


This part is a detailed edition of the last graph. These three pictures can verify the results we gained from the last part, because the works of every author show similar tendency. And they also provide some proofs that EAP nd MWS don’t like to use without in comparison with HPL.

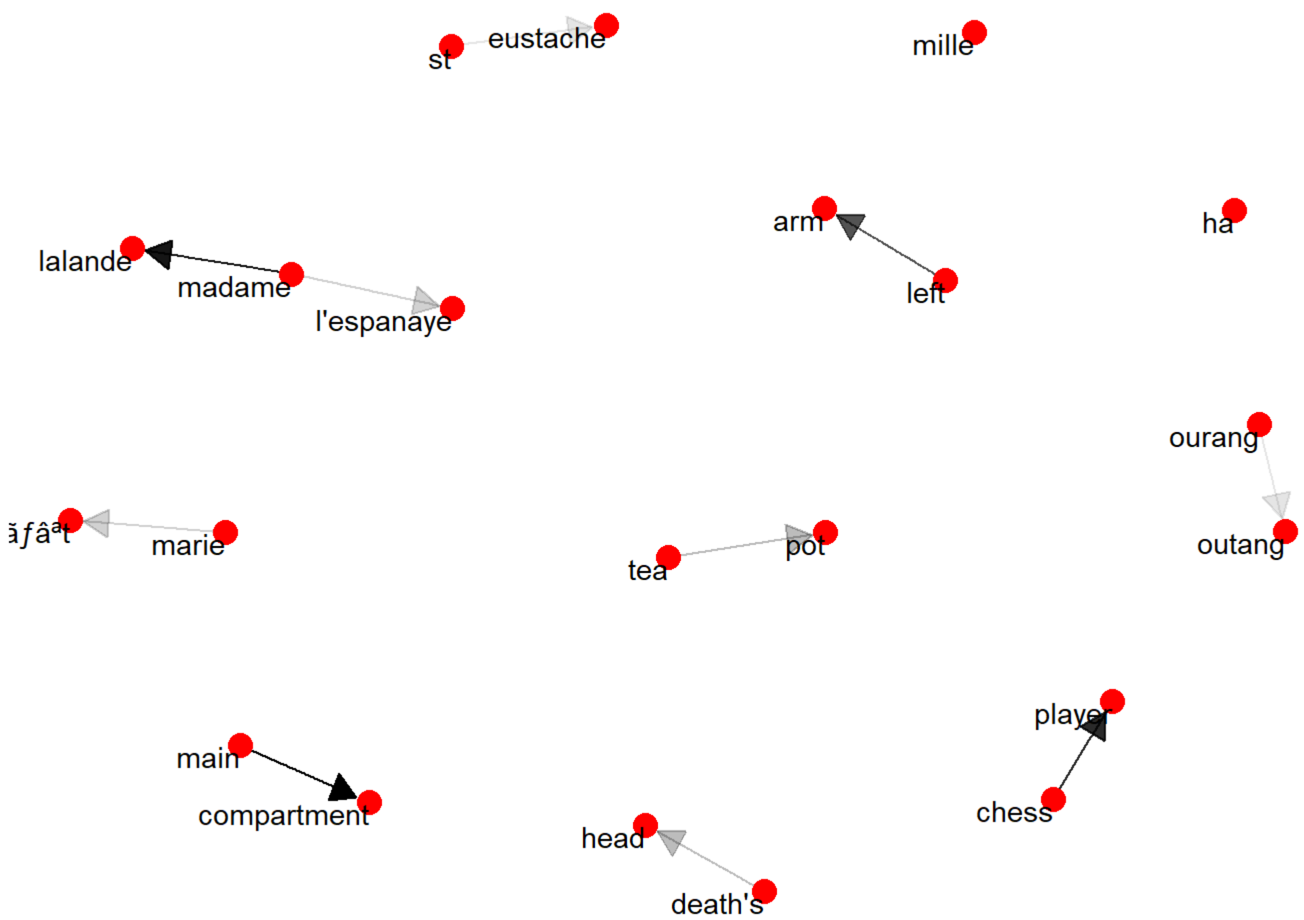
```
library(igraph)
bigram_graph <- bigram_counts %>%
  filter(n > 10) %>%
  graph_from_data_frame()

bigram_graph
```

```
## IGRAPH 870f203 DN-- 56 31 --
## + attr: name (v/c), n (e/n)
## + edges from 870f203 (vertex names):
##  [1] lord    ->raymond     fellow ->creatures   ha      ->ha
##  [4] main    ->compartment madame ->lalande     chess   ->player
##  [7] short   ->time        heh    ->heh         blue    ->eyes
## [10] left    ->arm         shunned->house       native  ->country
## [13] tempest->mountain     brown  ->jenkin      herbert->west
## [16] tea     ->pot         ten    ->thousand    death's->head
## [19] human   ->nature      human  ->race        marie   ->rogã□âªt
## [22] yog     ->sothoth     ancient->house       dr      ->armitage
## + ... omitted several edges
```

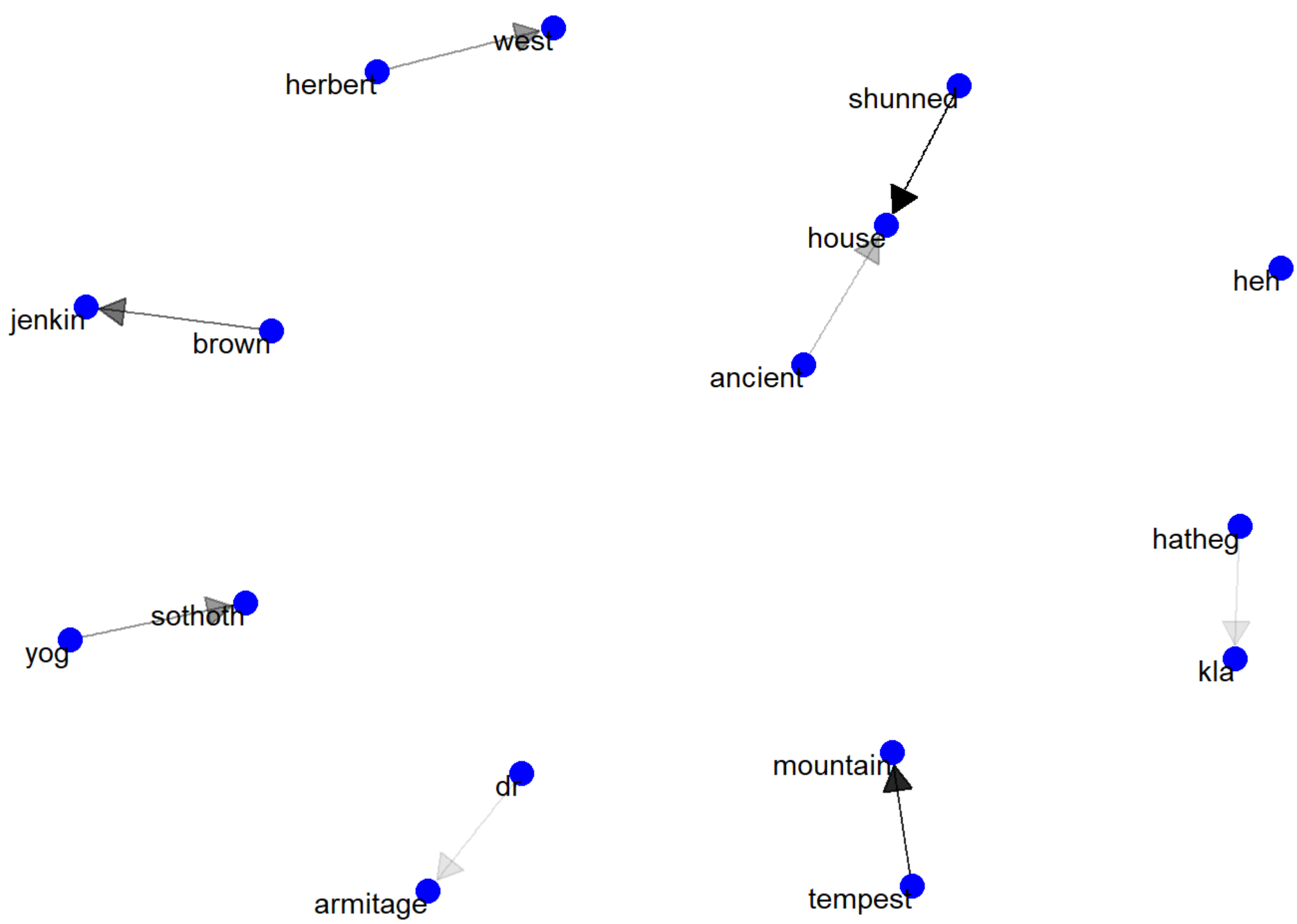
```
set.seed(2016)
library(ggraph)
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
#png("../figs/bigraph.png")
ggraph(bigram_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "purple", size = 4) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```



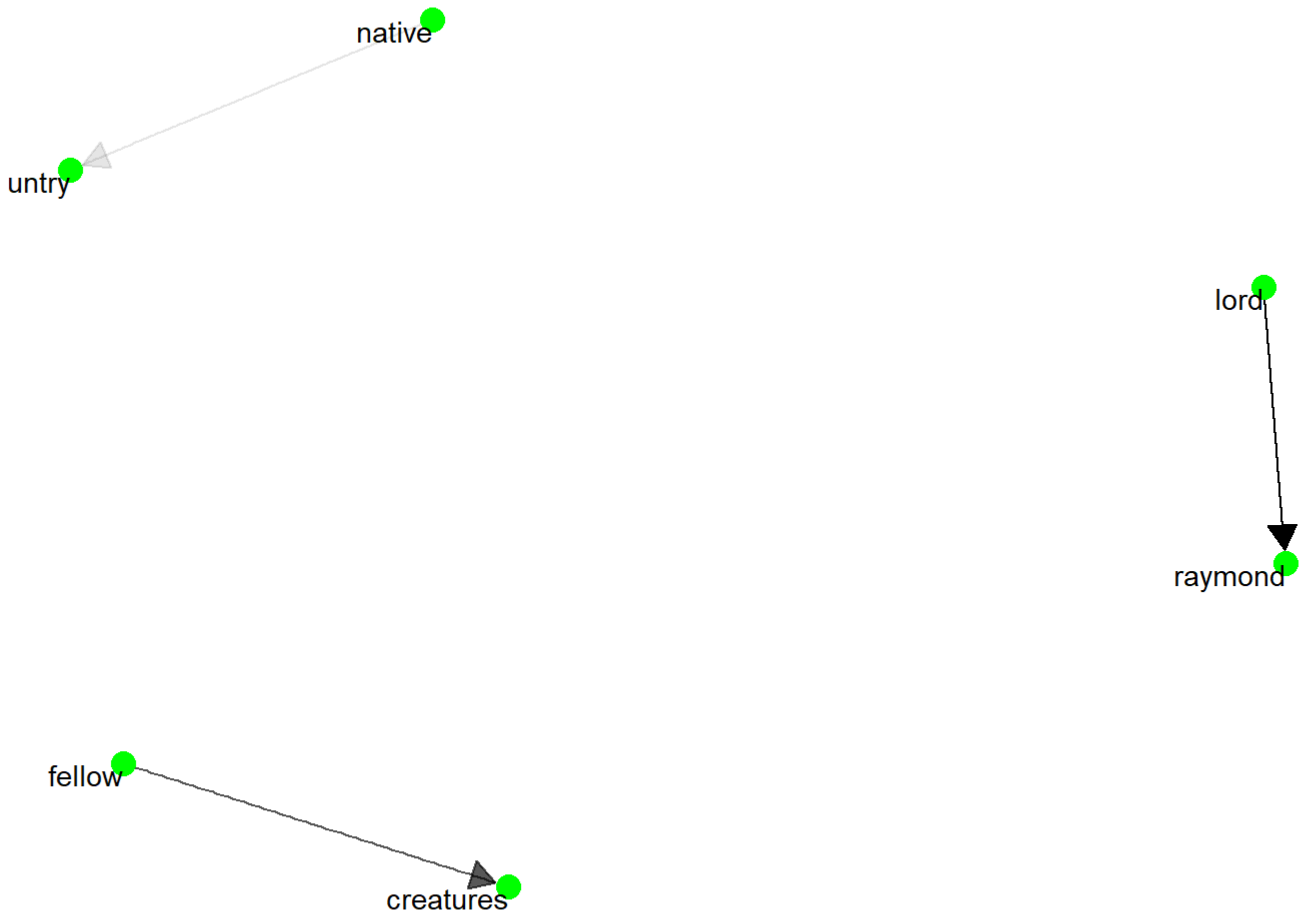
```
bigrams_eapsep <- eap_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_eapfilt <- bigrams_eapsep %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigram_countseap <- bigrams_eapfilt %>%
  count(word1, word2, sort = TRUE)

bigram_grapheap <- bigram_countseap %>%
  filter(n > 10) %>%
  graph_from_data_frame()

set.seed(2016)
library(ggraph)
#png("../figs/bigrapheap.png")
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
ggraph(bigram_grapheap, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "red", size = 4) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```



```
bigrams_hplsep <- hpl_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_hplfilt <- bigrams_hplsep %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigram_countshpl <- bigrams_hplfilt %>%
  count(word1, word2, sort = TRUE)

bigram_graphhpl <- bigram_countshpl %>%
  filter(n > 10) %>%
  graph_from_data_frame()
set.seed(2016)
library(ggraph)
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
#png("../figs/bigraphhpl.png")
ggraph(bigram_graphhpl, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "blue", size = 4) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```



```
bigrams_mwssep <- mws_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_mwsfilt <- bigrams_mwssep %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigram_countsmws <- bigrams_mwsfilt %>%
  count(word1, word2, sort = TRUE)

bigram_graphmws <- bigram_countsmws %>%
  filter(n > 10) %>%
  graph_from_data_frame()

set.seed(2016)
library(ggraph)
a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
#png("../figs/bigraphmws.png")
ggraph(bigram_graphmws, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "green", size = 4) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void()
```



Here are the graphs show the most popular bigrams htroughout the dataset, and the contribution from different authors. Arrows represent the inner directions of bigrams. The first graph discribes the total dataset, the second one is for EAP, the third one is for HPL. the forth one is for MWS. From the comparison, we can see EAP uses the most fixed bigrams, while MWS uses the least. But this may also because the discrepancies between total quantities of words from different authors. Besides, EAP mentions more daily supplies, HPL mentions more locations, while MWS mentions more nouns about a country.

8. Topic Models We use the `topicmodels` package for this analysis. Since the `topicmodels` package doesn't use the `tidytext` framework, we first convert our `spooky_wrd` dataframe into a document term matrix (DTM) matrix using `tidytext` tools.

```
# Counts how many times each word appears in each sentence
spk_wrd <- unnest_tokens(spk, word, text)
spk_wrd <- anti_join(spk_wrd, stop_words, by = "word")
swrd_freqs <- count(spk_wrd, id, word)
head(swrd_freqs)
```

```
## # A tibble: 6 x 3
##   id      word         n
##   <chr>   <chr>    <int>
## 1 id00001 content      1
## 2 id00001 idris        1
## 3 id00001 mine         1
## 4 id00001 resolve      1
## 5 id00002 accursed     1
## 6 id00002 city         1
```

```
# Creates a DTM matrix
spk_wrd_tm <- cast_dtm(swrd_freqs, id, word, n)
spk_wrd_tm
```

```
## <<DocumentTermMatrix (documents: 19467, terms: 24957)>>
## Non-/sparse entries: 194023/485643896
## Sparsity           : 100%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

```
length(unique(spk_wrd$id))
```

```
## [1] 19467
```

```
length(unique(spk_wrd$word))
```

```
## [1] 24957
```

The matrix `spooky_wrd_tm` is a sparse matrix with 19467 rows, corresponding to the 19467 ids (or originally, sentences) in the `spooky_wrd` dataframe, and 24941 columns corresponding to the total number of unique words in the `spooky_wrd` dataframe. So each row of `spooky_wrd_tm` corresponds to one of the original sentences. The value of the matrix at a certain position is then the number of occurences of that word (determined by the column) in this specific sentence (determined by the row). Since most sentence/word pairings don't occur, the matrix is sparse meaning there are many zeros.

For LDA we must pick the number of possible topics. Let's try 12, though this selection is admittedly arbitrary.

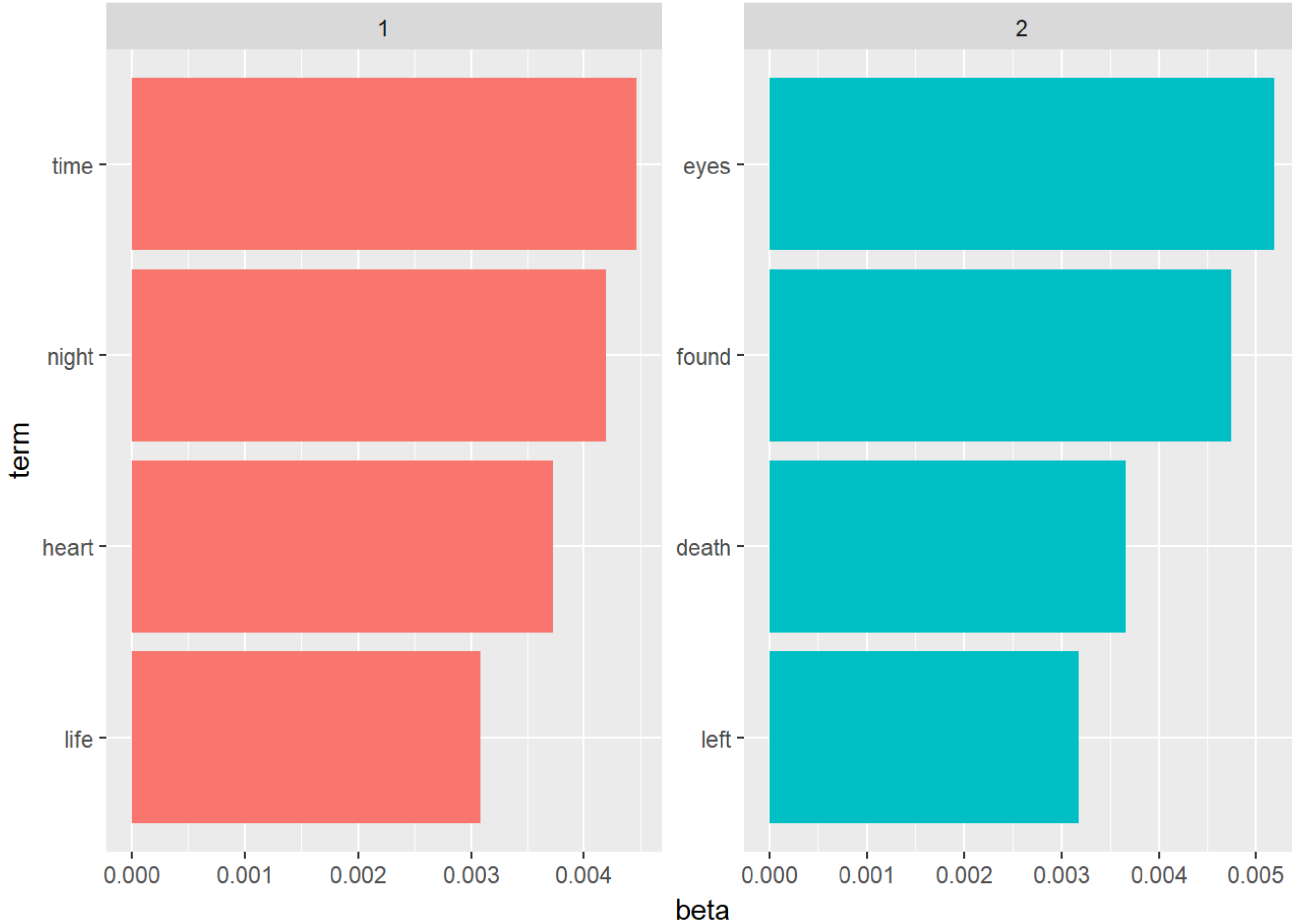
```
spk_wrd_lda <- LDA(spk_wrd_tm, k = 2, control = list(seed = 1234))
spk_wrd_top <- tidy(spk_wrd_lda, matrix = "beta")
spk_wrd_top
```

```
## # A tibble: 49,914 x 3
##    topic term          beta
##    <int> <chr>        <dbl>
##  1     1 content  0.000185
##  2     2 content  0.000159
##  3     1 idris    0.000377
##  4     2 idris    0.000725
##  5     1 mine     0.000637
##  6     2 mine     0.000323
##  7     1 resolve  0.0000469
##  8     2 resolve  0.000105
##  9     1 accursed 0.000219
## 10     2 accursed 0.000186
## # ... with 49,904 more rows
```

Topics Terms

We note that in the above we use the `tidy` function to extract the per-topic-per-word probabilities, called "beta" or β, for the model. The final output has a one-topic-per-term-per-row format. For each combination, the model computes the probability of that term being generated from that topic. For example, the term "content" has a 1.619628×10^{-5} probability of being generated from topic 4. We visualize the top terms (meaning the most likely terms associated with each topic) in the following.

```
# Grab the top five words for each topic.
spk_wrd_top_4 <- ungroup(top_n(group_by(spk_wrd_top, topic), 4, beta))
spk_wrd_top_4 <- arrange(spk_wrd_top_4, topic, -beta)
spk_wrd_top_4 <- mutate(spk_wrd_top_4, term = reorder(term, beta))

ggplot(spk_wrd_top_4) +
  geom_col(aes(term, beta, fill = factor(topic)), show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol = 3) +
  coord_flip()
```



In the above, we see that the first topic is characterized by words like “love”, “earth”, and “words” while the third topic includes the word “thousand”, and the fifth topic the word “beauty”. Note that the words “eyes” and “time” appear in many topics. This is the advantage to topic modelling as opposed to clustering when using natural language – often a word may be likely to appear in documents characterized by multiple topics.

We can also study terms that have the greatest difference in probabilities between the topics, ignoring the words that are shared with similar frequency between topics. We choose only the first 3 topics as example and visualise the differences by plotting log ratios: $log_{10}(\beta \text{ of topic x } / \beta \text{ of topic y})$. So if a word is 10 times more frequent in topic x the log ratio will be 1, whereas it will be -1 if the word is 10 times more frequent in topic y.

```
beta_spread_12 <- spk_wrd_top %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1 > .001 | topic2 > .001) %>%
  mutate(log_ratio = log2(topic2 / topic1))

beta_spread_12 <- group_by(beta_spread_12, direction = log_ratio > 0)
beta_spread_12 <- ungroup(top_n(beta_spread_12, 5, abs(log_ratio)))
beta_spread_12 <- mutate(beta_spread_12, term = reorder(term, log_ratio))

lr12 <- ggplot(beta_spread_12) +
      geom_col(aes(term, log_ratio, fill = log_ratio > 0)) +
      theme(legend.position = "none") +
      labs(y = "Log ratio of beta in topic 2 / topic 1") +
      coord_flip()

beta_spread_13 <- spk_wrd_top %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1 > .001 | topic2 > .001) %>%
  mutate(log_ratio = log2(topic2 / topic1))

beta_spread_13 <- group_by(beta_spread_13, direction = log_ratio > 0)
beta_spread_13 <- ungroup(top_n(beta_spread_13, 5, abs(log_ratio)))
beta_spread_13 <- mutate(beta_spread_13, term = reorder(term, log_ratio))

lr13 <- ggplot(beta_spread_13) +
      geom_col(aes(term, log_ratio, fill = log_ratio > 0)) +
      theme(legend.position = "none") +
      labs(y = "Log ratio of beta in topic 3 / topic 1") +
      coord_flip()

beta_spread_23 <- spk_wrd_top %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1 > .001 | topic2 > .001) %>%
  mutate(log_ratio = log2(topic2 / topic1))

beta_spread_23 <- group_by(beta_spread_23, direction = log_ratio > 0)
beta_spread_23 <- ungroup(top_n(beta_spread_23, 5, abs(log_ratio)))
beta_spread_23 <- mutate(beta_spread_23, term = reorder(term, log_ratio))

lr23 <- ggplot(beta_spread_23) +
      geom_col(aes(term, log_ratio, fill = log_ratio > 0)) +
      theme(legend.position = "none") +
      labs(y = "Log ratio of beta in topic 3 / topic 2") +
      coord_flip()
```

In the above, the words more common to topic 2 than topic 1 are “moon”, “air”, and “window” while the words more common to topic 1 are “moment”, “marie”, and “held”.

Sentence Topics

Above we look at the words representing each topic, we can also study the topics representing each documents, or in our case sentence. We use the `tidy` function to extract the per-document-per-topic probabilities, called "gamma" or γ, for the model.

```
spk_wrd_docs <- tidy(spk_wrd_lda, matrix = "gamma")
spk_wrd_docs
```

```
## # A tibble: 38,934 x 3
##    document topic gamma
##    <chr>    <int> <dbl>
##  1 id00001      1 0.498
##  2 id00002      1 0.519
##  3 id00003      1 0.496
##  4 id00004      1 0.501
##  5 id00005      1 0.487
##  6 id00006      1 0.507
##  7 id00007      1 0.504
##  8 id00009      1 0.537
##  9 id00010      1 0.494
## 10 id00012      1 0.497
## # ... with 38,924 more rows
```

The above table holds the estimated proportion of words from that sentence (id) that are generated from that topic. For example, the model estimates that only about 8.301% of the words in sentence id00001 were generated from topic 1.

```
author_top <- left_join(spk_wrd_docs, spk, by = c("document" = "id"))
author_top <- select(author_top, -text)
author_top$topic <- as.factor(author_top$topic)

# Chooses the top topic per sentence
author_top <- ungroup(top_n(group_by(author_top, document), 1, gamma))

# Counts the number of sentences represented by each topic per author
author_top <- ungroup(count(group_by(author_top, author, topic)))

author_top
```

```
## # A tibble: 6 x 3
##   author topic     n
##   <fct>  <fct> <int>
## 1 EAP    1      3902
## 2 EAP    2      3937
## 3 HPL    1      2735
## 4 HPL    2      2874
## 5 MWS    1      3089
## 6 MWS    2      2930
```

```
ggplot(author_top) +
  geom_col(aes(topic, n, fill = factor(topic)), show.legend = FALSE) +
  facet_wrap(~ author, scales = "free", ncol = 4) +
  coord_flip()
```