Spooky Text Analysis

Richard Shin

Introduction

In class, we looked at a dataset, spooky.csv, which contains excerpts of texts from Edgar Allan Poe, H.P. Lovecraft, and Mary Shelley. The analysis of the dataset included using the tidytext library to clean up the data, and then undergoing sentiment analysis as well as topic modeling. This

Libraries

```r
library(ggplot2);library(dplyr);library(tibble)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr);library(stringr);library(tidytext)
library(topicmodels);library(wordcloud);library(ggridges)
```

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

Data Overview

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

```
##        id
## 1 id26305
## 2 id17569
## 3 id11008
## 4 id27763
## 5 id12958
## 6 id22965
##
## 1
## 2
## 3
## 4
## 5
## 6 A youth passed in solitude, my best years spent under your gentle and feminine fosterage, has so re
##   author
## 1    EAP
```

```
## 2    HPL
## 3    EAP
## 4    MWS
## 5    HPL
## 6    MWS
```

```r
summary(spooky)
```

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

```r
spooky$author <- as.factor(spooky$author)
```

Methods

Verify that there are no missing values in the entire dataset.

```r
sum(is.na(spooky))
```

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

Initial goal: check the top frequency of how much each stop word is used, and see if we should keep some of them in. Only check stop words that are longer than 5 characters.

```r
spooky_wrd <- unnest_tokens(spooky, word, text)
head(spooky_wrd)
```

```
##          id author     word
## 1   id26305    EAP     this
## 1.1 id26305    EAP  process
## 1.2 id26305    EAP  however
## 1.3 id26305    EAP afforded
## 1.4 id26305    EAP       me
## 1.5 id26305    EAP       no
```

```r
head(stop_words)
```

```
## # A tibble: 6 x 2
##        word lexicon
##       <chr>   <chr>
## 1         a   SMART
## 2       a's   SMART
## 3      able   SMART
## 4     about   SMART
## 5     above   SMART
## 6 according   SMART
```

```r
freq_stop = vector()
stop_words_long= vector()
for (i in 1:nrow(stop_words)){
      if (nchar(stop_words[i,1])>5){
          stop_words_long = rbind(stop_words_long, stop_words[i,1])
      }
}
```

```r
head(stop_words_long)
```

```
## # A tibble: 6 x 1
##          word
##         <chr>
## 1   according
## 2 accordingly
## 3      across
## 4    actually
## 5  afterwards
## 6     against
```

```r
summary(stop_words_long)
```

```
##      word
##  Length:432
##  Class :character
##  Mode  :character
```

```r
spooky_wrd_long <- NULL
for (i in 1:nrow(spooky_wrd)){
      if (nchar(spooky_wrd[i,3])>5){
          spooky_wrd_long = rbind(spooky_wrd_long, spooky_wrd[i,3])
      }
}
head(spooky_wrd_long)
```

```
##      [,1]
## [1,] "process"
## [2,] "however"
## [3,] "afforded"
## [4,] "ascertaining"
## [5,] "dimensions"
## [6,] "dungeon"
```

```r
summary(spooky_wrd_long)
```

```
##        V1
##  before :   786
##  through:   586
##  should :   551
##  seemed :   544
##  little :   531
##  myself :   515
##  (Other):146790
```

Stop Word Frequency

```r
# stop_words_long$freq<- rep(0,nrow(stop_words_long))
# nxt <- 0
# for (i in 1:nrow(stop_words_long)){
#   for(j in 1:nrow(spooky_wrd_long)){
#     if (stop_words_long[i,1] == spooky_wrd_long[j]){
#       stop_words_long[i,2] = stop_words_long[i,2] + 1
```

```r
#       }
#   }
# }
```

This code chunk is too computationally intensive. Instead, will just cut out words from stop_words table with length greater than 5

```r
stop_words_short <- vector()
for (i in 1:nrow(stop_words)){
      if (nchar(stop_words[i,1])<5){
          stop_words_short = rbind(stop_words_short, stop_words[i,1])
      }
}
head(stop_words_short)
```

```
## # A tibble: 6 x 1
##    word
##   <chr>
## 1     a
## 2   a's
## 3  able
## 4   all
## 5  also
## 6    am
```

```r
summary(stop_words_short)
```

```
##      word
##  Length:501
##  Class :character
##  Mode  :character
```

Proceed with data cleaning

```r
spooky_wrd_new <- anti_join(spooky_wrd, stop_words_short, by = "word")
spooky_wrd_old <- anti_join(spooky_wrd, stop_words, by = "word")
summary(spooky_wrd_new)
```

```
##       id              author          word
##  Length:244249      EAP:91088   Length:244249
##  Class :character   HPL:76942   Class :character
##  Mode  :character   MWS:76219   Mode  :character
```

```r
summary(spooky_wrd)
```

```
##       id              author           word
##  Length:522818      EAP:200855   Length:522818
##  Class :character   HPL:156263   Class :character
##  Mode  :character   MWS:165700   Mode  :character
```

Wordcloud

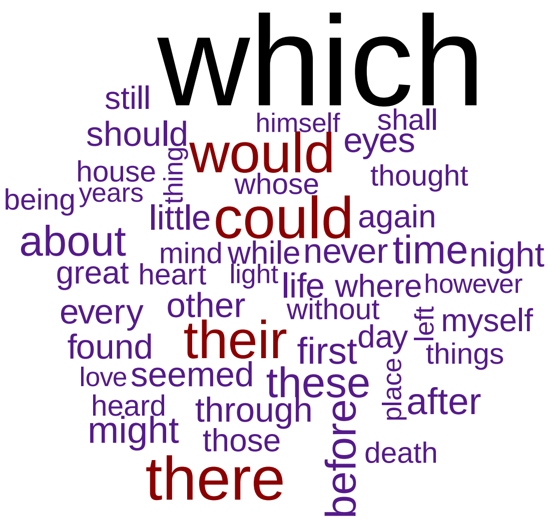
Create a wordcloud from the new spooky_wrd table using the updated stop words.

```r
# Words is a list of words, and freqs their frequencies
words <- count(group_by(spooky_wrd_new, word))$word
freqs <- count(group_by(spooky_wrd_new, word))$n
```

```r
head(sort(freqs, decreasing = TRUE))
```

```
## [1] 3369 1339 1316 1241 1160  788
```

```r
wordcloud(words, freqs, max.words = 50, color = c("purple4", "red4", "black"))
```



Many words which seem to be associated with a sense of doubt such as could, would, should. Some of the other biggest words in the cloud are also a stop word in the original set, such as "which," "there," and "their." Because of their high frequency and relative lack of meaning, "which," "there," and "their" will be added to the stop_words_short set and the word cloud will be remade.

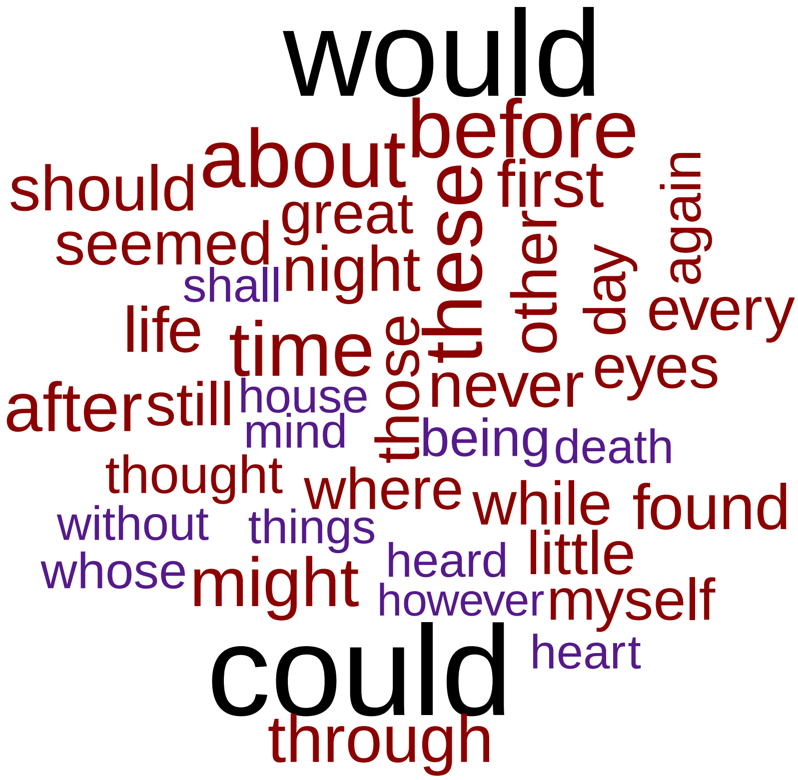
```r
new1 <- "which"
new2 <- "there"
new3 <- "their"
stop_words_short<- rbind(stop_words_short, new1)
stop_words_short<- rbind(stop_words_short, new2)
stop_words_short<- rbind(stop_words_short, new3)
tail(stop_words_short, 4)
```

```
## # A tibble: 4 x 1
##    word
##   <chr>
## 1  your
## 2 which
## 3 there
## 4 their
```

```r
spooky_wrd_new <- anti_join(spooky_wrd, stop_words_short, by = "word")
# Words is a list of words, and freqs their frequencies
words <- count(group_by(spooky_wrd_new, word))$word
freqs <- count(group_by(spooky_wrd_new, word))$n

head(sort(freqs, decreasing = TRUE))
```

```
## [1] 1316 1241  788  786  769  729
```

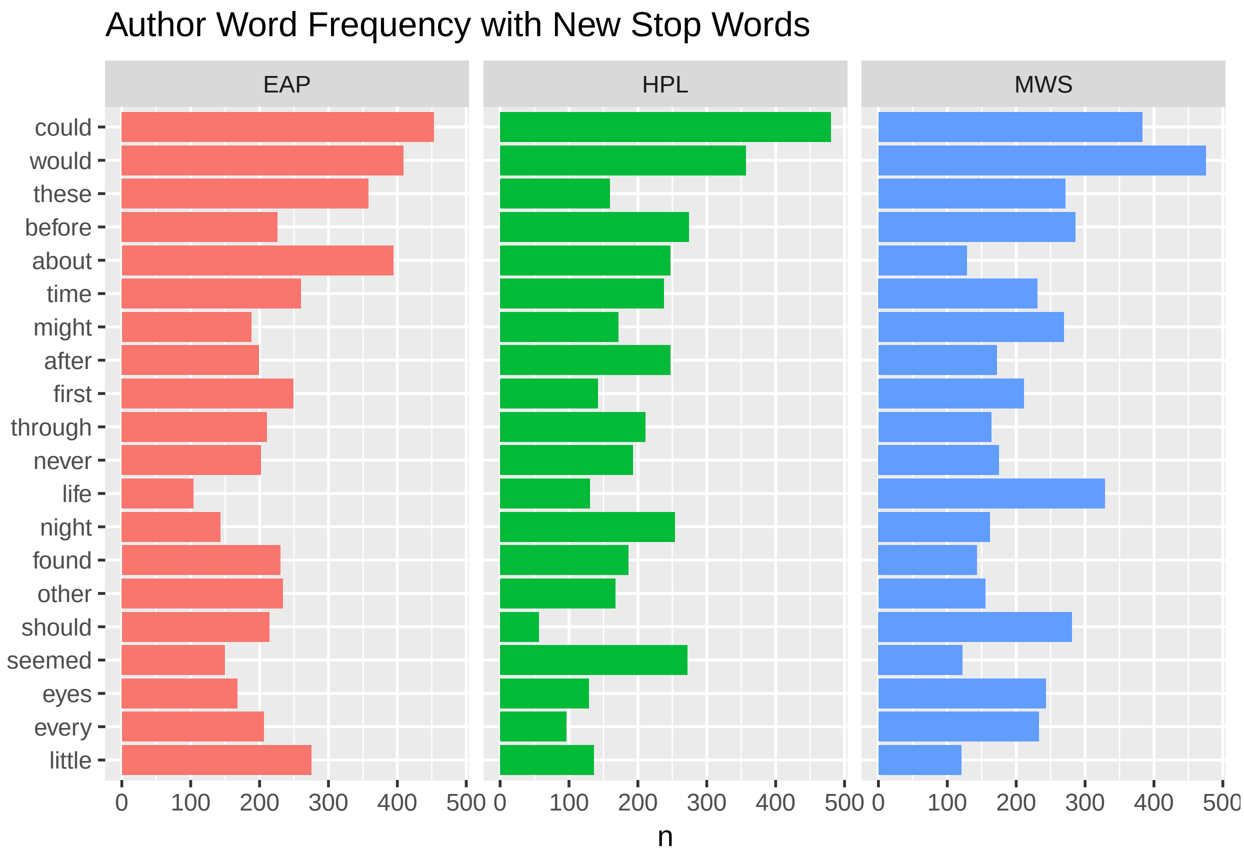
```r
wordcloud(words, freqs, max.words = 40, color = c("purple4", "red4", "black"))
```



Word Frequency

To put this wordcloud into perspective, the top words for each author including the former stop words will be displayed in a graph.

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

# Counts number of times each word was used.
all_words    <- rename(count(group_by(spooky_wrd_new, 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, 60))

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") +
  ggtitle("Author Word Frequency with New Stop Words")
```



What we see from these graphs is that the authors use conditional words such as "could" and "would" very often. Shelley uses "would" with hgiher frequency than "could" than Poe or Lovecraft. Other words which were not easily noticed in the word cloud are "might," "about," and "seemed," more words for uncertainty. While the authors appear to use "might" with relatively similar frequencies, for the other words they differ in their usage. In particular, Poe uses "about" with greater frequency than Lovecraft and Shelley. Meanwhile, Lovecraft uses "seemed" at nearly double the frequency of Poe and Shelley.

Sentiment Analysis

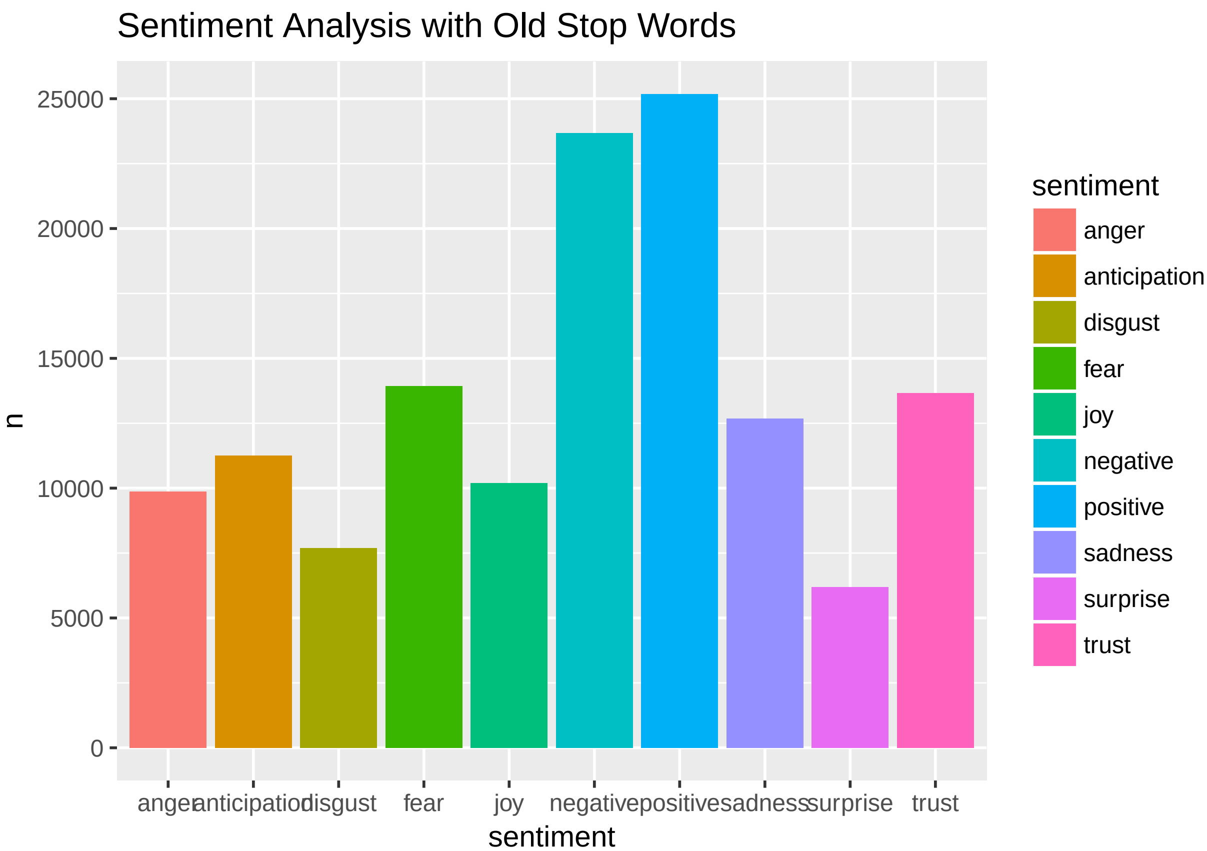
With this large frequency of words which purvey a sense of doubt and conditionality, perhaps sentiment analysis will proceed differently.

```r
#Old Stop Words Graph
get_sentiments('nrc')
```

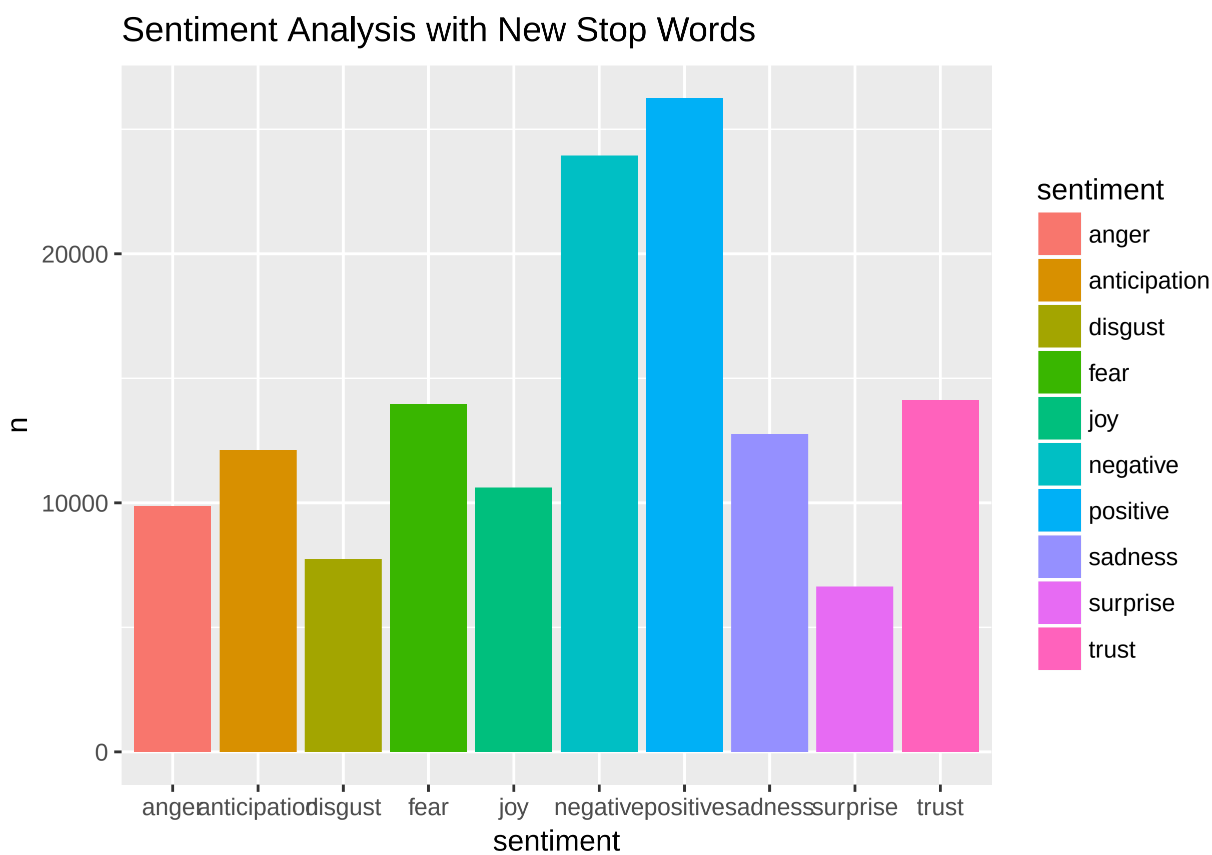
```
## # A tibble: 13,901 x 2
##           word sentiment
##          <chr>     <chr>
##  1      abacus     trust
##  2     abandon      fear
##  3     abandon  negative
##  4     abandon   sadness
##  5   abandoned     anger
##  6   abandoned      fear
##  7   abandoned  negative
##  8   abandoned   sadness
##  9 abandonment     anger
```

```
## 10 abandonment      fear
## # ... with 13,891 more rows
```

```r
sentiments <- inner_join(spooky_wrd_old, get_sentiments('nrc'), by = "word")
ggplot(count(sentiments, sentiment)) +
  geom_col(aes(sentiment, n, fill = sentiment)) +
  ggtitle("Sentiment Analysis with Old Stop Words")
```




```r
#New Stop Words Graph
sentiments_new <- inner_join(spooky_wrd_new, get_sentiments('nrc'), by = "word")
ggplot(count(sentiments_new["sentiment"], sentiment)) +
  geom_col(aes(sentiment, n, fill = sentiment)) +
  ggtitle("Sentiment Analysis with New Stop Words")
```




```r
count(sentiments, sentiment)
```

```
## # A tibble: 10 x 2
##       sentiment     n
##           <chr> <int>
##  1        anger  9869
##  2 anticipation 11258
##  3      disgust  7697
##  4         fear 13927
##  5          joy 10190
##  6     negative 23674
##  7     positive 25175
##  8      sadness 12674
##  9     surprise  6199
## 10        trust 13655
```

```r
count(sentiments_new, sentiment)
```

```
## # A tibble: 10 x 2
##       sentiment     n
##           <chr> <int>
##  1        anger  9869
##  2 anticipation 12124
##  3      disgust  7731
##  4         fear 13960
##  5          joy 10615
##  6     negative 23948
##  7     positive 26246
```

```
## 8      sadness 12760
## 9     surprise  6626
## 10       trust 14126
```

```
#Sentiments in order: anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, t
count(sentiments_new, sentiment)$n - count(sentiments, sentiment)$n
```

```
## [1]    0  866   34   33  425  274 1071   86  427  471
```

```
(count(sentiments_new, sentiment)$n - count(sentiments, sentiment)$n)/count(sentiments, sentiment)$n
```

```
## [1] 0.000000000 0.076923077 0.004417305 0.002369498 0.041707556
## [6] 0.011573879 0.042542205 0.006785545 0.068882078 0.034492860
```

It is hard to tell from the sheer number of data points on these graphs, but by comparing the counts, the numbers for the new stop words sets is, as expected, greater in count than the old set. Particularly, for anticipation, and, surprisingly, positive, there are 800 more occurrences of these words in the new set than in the old. However, in terms of relative increase, anticipation goes up by 7.7%, while positive only goes up by 4.3%. Another notable increase is in sadness and surprise, which increased by 6.8 and 6.9% respectively. This suggests that either there are many long words in the original stop words data set that pertain to anticipation, sadness, and surprise which were used by the authors, or that the authors often used these longer words in their stories.

Lexicon Analysis

Perhaps the lexicon should be inspected. We will check the most common former stop words, "could," "would," "might," "about," and "seemed."

```
lex <- get_sentiments("nrc")
summary(lex)
```

```
##      word            sentiment
##  Length:13901       Length:13901
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
lex[lex$word == "could",2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "would", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "might", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "about", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "seemed", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

Not surprisingly, these words lack sentiments as they can be used in a variety of contexts. This means that these words, which take up a huge proportion of the total words used by these authors, are not contributing to the net increase in anticipation, sadness, and surprise. This means that the authors are most likely using several different stop words which have these sentiments, but overall do not repeat them very much.

TF-IDF

To investigate the usage of low frequency stop words being used, we will use TF-IDF analysis to see if there are any words characteristic to each author.

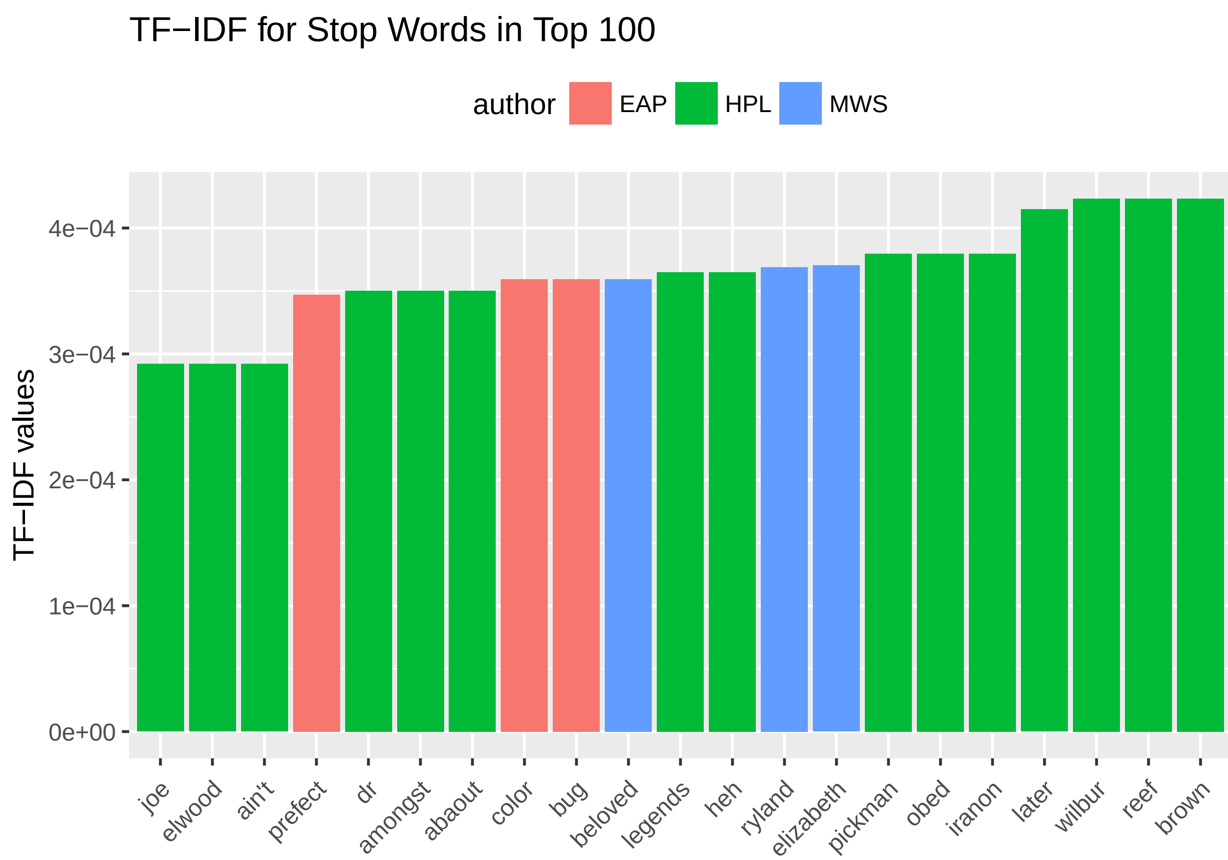
```
#do tf-idf for the old word list
frequency <- count(spooky_wrd_old, author, word)
tf_idf    <- bind_tf_idf(frequency, word, author, n)
tf_idf    <- arrange(tf_idf, desc(tf_idf))
tf_idf    <- mutate(tf_idf, word = factor(word, levels = rev(unique(word))))

# Grab the top one-hundred tf_idf scores in all the words and omit the top 20 to account for names bein
tf_idf_100 <- top_n(tf_idf, 100, tf_idf)
tf_idf_100 <- tf_idf_100[-c(1:20),]


#do the same for the new word list
frequency <- count(spooky_wrd_new, author, word)
tf_idf    <- bind_tf_idf(frequency, word, author, n)
tf_idf    <- arrange(tf_idf, desc(tf_idf))
tf_idf    <- mutate(tf_idf, word = factor(word, levels = rev(unique(word))))


tf_idf_100_new <- top_n(tf_idf, 100, tf_idf)
tf_idf_100_new <- tf_idf_100_new[-c(1:20),]

#Now find the difference between the two
diff <- tf_idf_100_new$word[!tf_idf_100_new$word %in% tf_idf_100$word]
diff
```

```
## [1] later   amongst ain't
## 25359 Levels: zest zeal youthful youth yourselves yourself yours ... perdita
```

```
c(tf_idf_100_new[tf_idf_100_new$word == diff[1],6],tf_idf_100_new[tf_idf_100_new$word == diff[2],6], tf
```

```
## $tf_idf
## [1] 0.000414867
##
## $tf_idf
## [1] 0.0003503647
##
## $tf_idf
## [1] 0.0002919706
```

```
ggplot(tf_idf_100_new[-c(1:6,11:15, 30:49,53:80),]) +
  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))+
  ggtitle("TF-IDF for Stop Words in Top 100 ")
```



In an inspection of the top 100 TF-IDF scored words, three of them turn out to be former stop words, and all three of these are characteristic of Lovecraft. While these words may not purvey the sense of doubt initially characteristic of the former stop words, perhaps we have found the branch of former stop words which contribute to the sentiment analysis.

```
lex[lex$word == "ain't", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "amongst", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

```
lex[lex$word == "later", 2]
```

```
## # A tibble: 0 x 1
## # ... with 1 variables: sentiment <chr>
```

Not quite. This means that the words from the stop table are buried deep in the texts, but are not quite exclusively used by each author to the point that the author could be identified.

Summary

The spooky data set was inspected after reevaluating the stop words data set. We found that there were many occurrences of words purveying doubt and uncertainty, but ultimately that the most frequent former stop words lacked a defined sentiment. This was strange, however, since the sentiment analysis showed vast increases in the frequency of words pertaining to anticipation and surprise. Thus, by inspecting the less-used former stop words with TF-IDF, we hoped to find that these words were related to anticipation or surprise, or really any sentiment, but once again came up with inconclusive results. The former stop words which contributed to the sentiment analysis remain undiscovered and obscure, perhaps a memento to the spooky stories which they are a part of.