(b): Analyzing bigrams

A bigram can also be treated as a term in a document in the same way that we treated individual words. For example, we can look at the tf-idf of bigrams across spooky dataset.

TF stands for term frequency or how often a word appears in a text and it is what is studied above in the word cloud. IDF stands for inverse document frequncy, and it is a way to pay more attention to words that are rare within the entire set of text data that is more sophisticated than simply removing stop words. Multiplying these two values together calculates a term's tf-idf, which is the frequency of a term adjusted for how rarely it is used. We'll use tf-idf as a heuristic index to indicate how frequently a certain author uses a word relative to the frequency that all the authors use the word. Therefore we will find words that are characteristic for a specific author, a good thing to have if we are interested in solving the author identification problem.

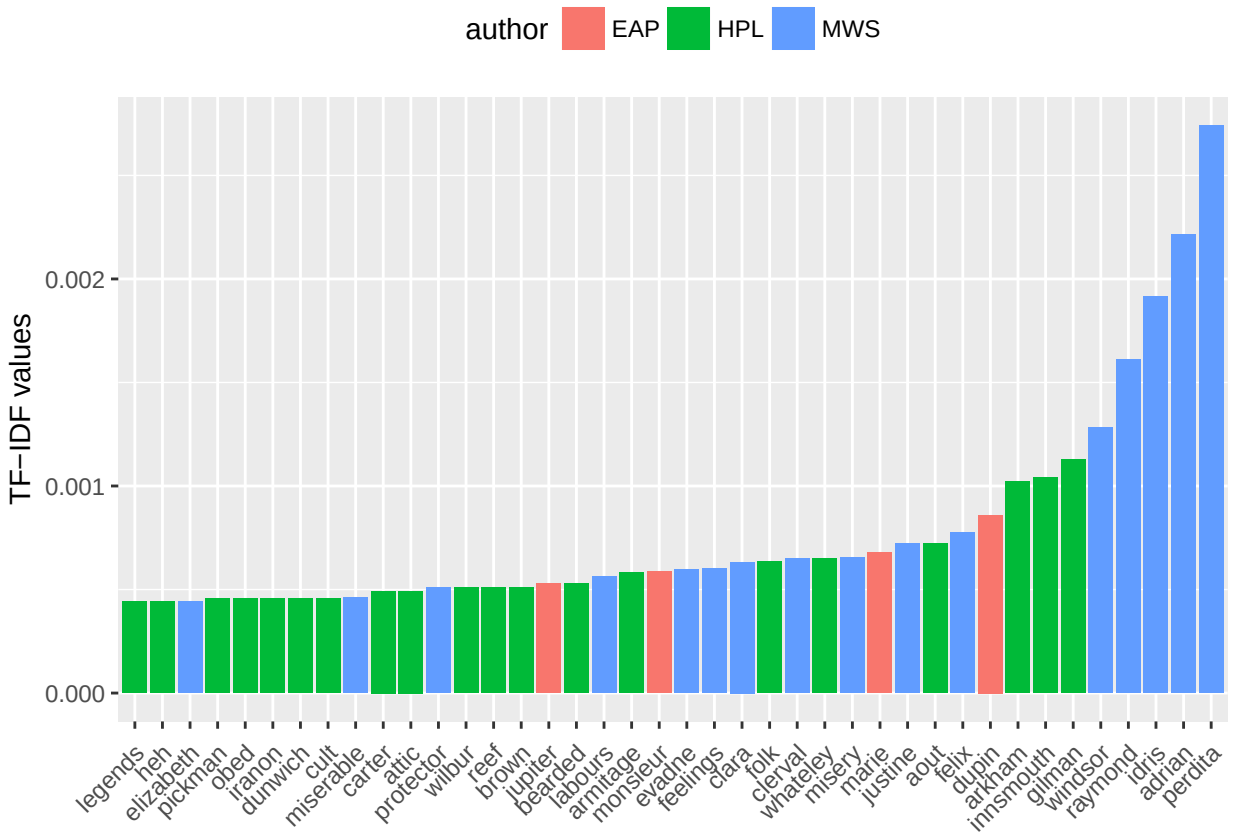
```
#get rid of stop words
spooky_wrd <- anti_join(spooky_wrd, stop_words, by = "word")
frequency<-count(spooky_wrd,author,word)
tf_idf<-bind_tf_idf(frequency,word,author,n)
head(tf_idf)
```

```
## # A tibble: 6 x 6
##   author word        n        tf   idf    tf_idf
##   <chr>  <chr>   <int>     <dbl> <dbl>     <dbl>
## 1 EAP    à           9 0.000124  1.10  0.000136
## 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 0.0000275 1.10  0.0000302
## 6 EAP    abandon     7 0.0000961 0     0
```

```
tail(tf_idf)
```

```
## # A tibble: 6 x 6
##   author word         n        tf   idf     tf_idf
##   <chr>  <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
```

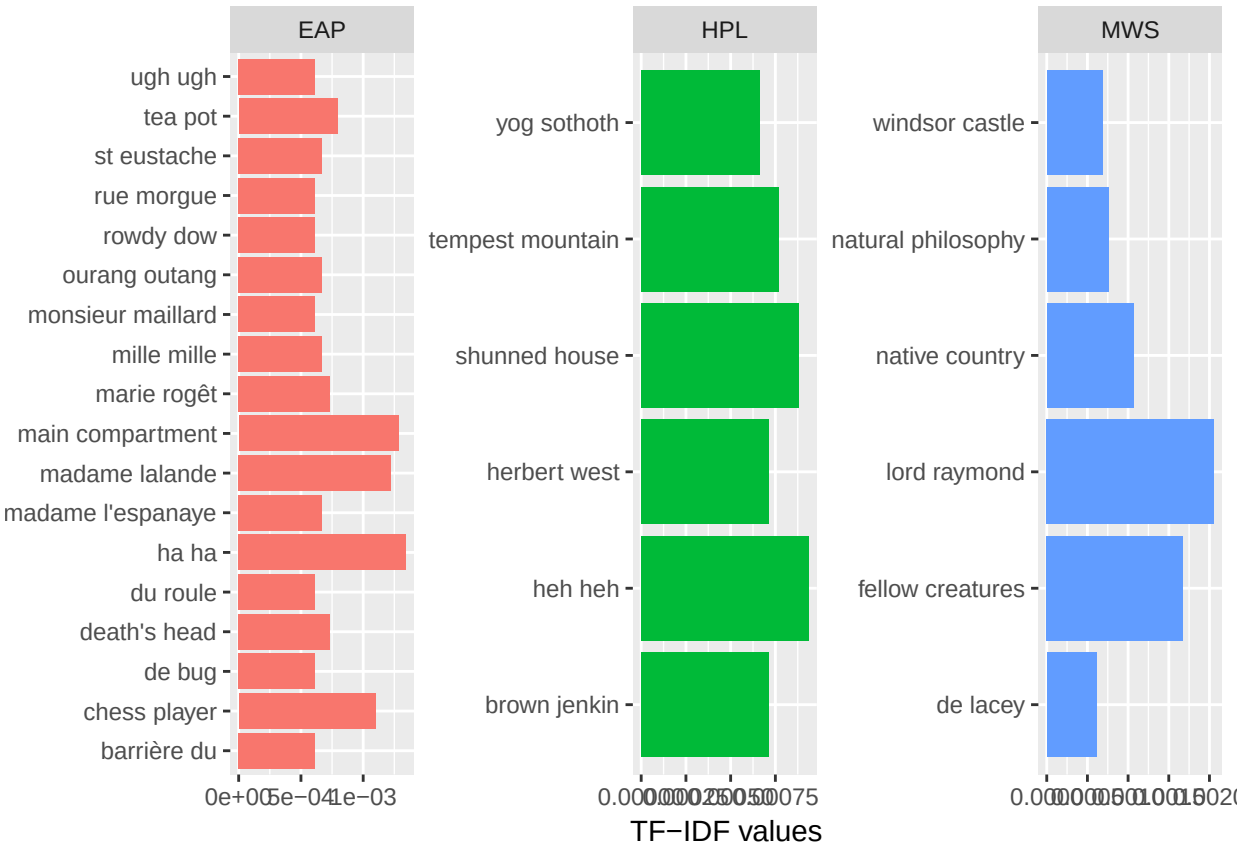
```
tf_idf<-arrange(tf_idf,desc(tf_idf))
tf_idf<-mutate(tf_idf, word = factor(word,levels= rev(unique(word))))

# Grab the top fourty tf_idf scores in all the words
tf_idf_40<- top_n(tf_idf,40,tf_idf)

ggplot(tf_idf_40) +
  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))
```



Note that in the above, many of the words recognized by their tf-idf scores are names. This makes sense – if we see text referencing Raymond, Idris, or Perdita, we know almost for sure that MWS is the author. But some non-names stand out. EAP often uses "monsieur" and "jupiter" while HPL uses the words "bearded" and "attic" more frequently than the others. We can also look at the most characteristic terms per author.

Then we can look at the tf-idf of bigrams across spooky datasts.

```
bigram_tf_idf<-bigrams_united %>%
  count(author,bigram) %>%
  bind_tf_idf(bigram,author,n) %>%
  arrange(desc(tf_idf))
bigram_tf_idf_30<-head(bigram_tf_idf,30)
ggplot(bigram_tf_idf_30) +
  geom_col(aes(bigram,tf_idf, fill = author)) +
  labs(x = NULL, y = "bigram_tf_idf") +
  theme(legend.position = "none") +
  facet_wrap(~ author,ncol =3,scales="free")+
  coord_flip() +
  labs(y = "TF-IDF values")
```



There are advantages and disadvantages to examining the tf-idf of bigrams rather than individual words. Pairs of consecutive words might capture structure that isn't present when one is just counting single words, and may provide context that makes tokens more understandable. However, the per-bigram counts are also sparser: a typical two-word pair is rarer than either of its component words.