“Not” isn’t the only term that provides some context for the following word. We could pick four common words (or more) that negate the subsequent term, and use the same joining and counting approach to examine all of them at once.

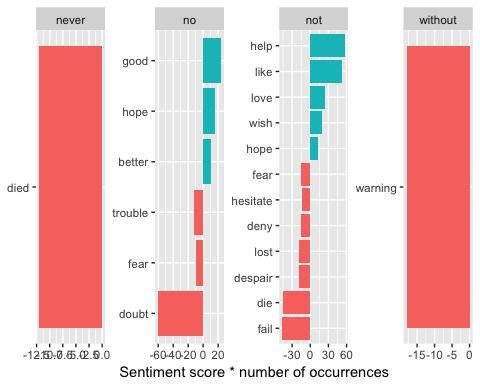
negation\_words <- c("not", "no", "never", "without")  
  
negated\_words<-bigrams\_separated %>%  
 filter(word1 %in% negation\_words) %>%  
 inner\_join(AFINN, by = c(word2 = "word")) %>%  
 count(word1, word2, score, sort = TRUE)   
head(negated\_words)

## # A tibble: 6 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

negated\_words$word1<-as.factor(negated\_words$word1)  
unique(negated\_words$word1)

## [1] no not never without  
## Levels: never no not without

negated\_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) +  
 labs(x = NULL, y = "Sentiment score \* number of occurrences")+  
 facet\_wrap(~ word1,ncol =4,scales="free")+  
 coord\_flip()

 “not doubt” and “not help” are the two most common examples, we can also see pairings such as “no hope” and “never forget.” We could combine this to reverse the AFINN scores of each word that follows a negation.

### 2: Compare Afinn, Bing with NRC

# Keep words that have been classified within the NRC lexicon.  
get\_sentiments('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

sentiments\_afinn <- inner\_join(spooky\_wrd, get\_sentiments('afinn'), by = "word")  
head(sentiments\_afinn)

## id author word score  
## 1 id26305 EAP no -1  
## 2 id26305 EAP perfectly 3  
## 3 id17569 HPL mistake -2  
## 4 id11008 EAP cutting -1  
## 5 id11008 EAP fantastic 4  
## 6 id11008 EAP greatest 3

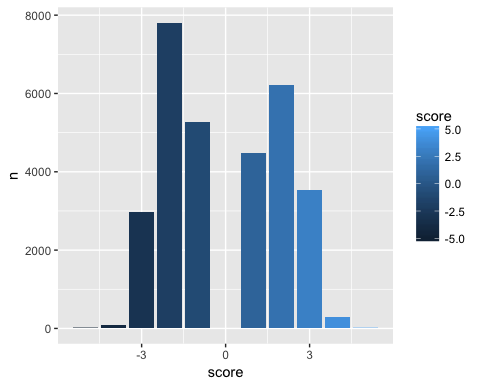
count(sentiments\_afinn, score)

## # A tibble: 10 x 2  
## score n  
## <int> <int>  
## 1 -5 12  
## 2 -4 98  
## 3 -3 2961  
## 4 -2 7810  
## 5 -1 5280  
## 6 1 4479  
## 7 2 6220  
## 8 3 3529  
## 9 4 291  
## 10 5 9

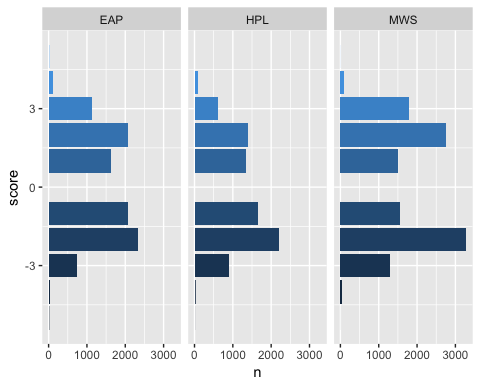
count(sentiments\_afinn, author, score)

## # A tibble: 29 x 3  
## author score n  
## <chr> <int> <int>  
## 1 EAP -5 9  
## 2 EAP -4 28  
## 3 EAP -3 751  
## 4 EAP -2 2321  
## 5 EAP -1 2072  
## 6 EAP 1 1639  
## 7 EAP 2 2071  
## 8 EAP 3 1121  
## 9 EAP 4 113  
## 10 EAP 5 7  
## # ... with 19 more rows

ggplot(count(sentiments\_afinn, score)) +   
 geom\_col(aes(score, n, fill = score))



ggplot(count(sentiments\_afinn, author, score)) +   
 geom\_col(aes(score, n, fill = score)) +   
 facet\_wrap(~ author) +  
 coord\_flip() +  
 theme(legend.position = "none")



get\_sentiments('bing')

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

sentiments\_bing<- inner\_join(spooky\_wrd, get\_sentiments('bing'), by = "word")  
head(sentiments\_bing)

## id author word sentiment  
## 1 id26305 EAP dungeon negative  
## 2 id26305 EAP perfectly positive  
## 3 id17569 HPL mistake negative  
## 4 id11008 EAP gold positive  
## 5 id11008 EAP fantastic positive  
## 6 id11008 EAP incessantly negative

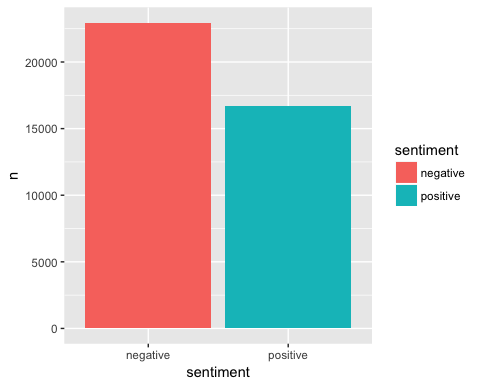
count(sentiments\_bing,sentiment)

## # A tibble: 2 x 2  
## sentiment n  
## <chr> <int>  
## 1 negative 22958  
## 2 positive 16674

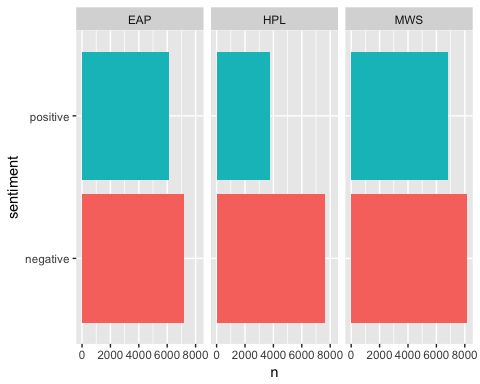
count(sentiments\_bing,author,sentiment)

## # A tibble: 6 x 3  
## author sentiment n  
## <chr> <chr> <int>  
## 1 EAP negative 7203  
## 2 EAP positive 6144  
## 3 HPL negative 7605  
## 4 HPL positive 3731  
## 5 MWS negative 8150  
## 6 MWS positive 6799

ggplot(count(sentiments\_bing,sentiment)) +   
 geom\_col(aes(sentiment, n, fill = sentiment))



ggplot(count(sentiments\_bing, author, sentiment)) +   
 geom\_col(aes(sentiment, n, fill = sentiment)) +   
 facet\_wrap(~ author) +  
 coord\_flip() +  
 theme(legend.position = "none")



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

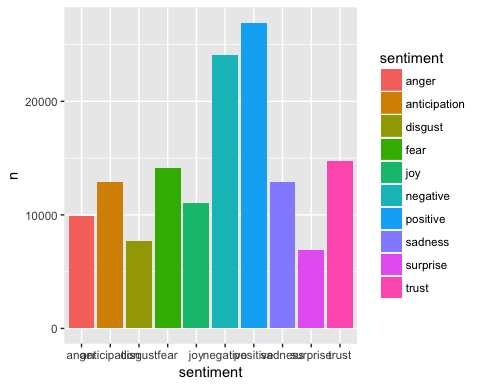
sentiments <- inner\_join(spooky\_wrd, get\_sentiments('nrc'), by = "word")  
  
count(sentiments, sentiment)

## # A tibble: 10 x 2  
## sentiment n  
## <chr> <int>  
## 1 anger 9869  
## 2 anticipation 12912  
## 3 disgust 7731  
## 4 fear 14096  
## 5 joy 11077  
## 6 negative 24084  
## 7 positive 26934  
## 8 sadness 12896  
## 9 surprise 6903  
## 10 trust 14777

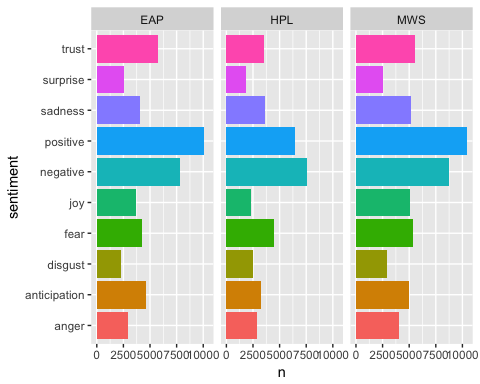
count(sentiments, author, sentiment)

## # A tibble: 30 x 3  
## author sentiment n  
## <chr> <chr> <int>  
## 1 EAP anger 2962  
## 2 EAP anticipation 4656  
## 3 EAP disgust 2273  
## 4 EAP fear 4287  
## 5 EAP joy 3652  
## 6 EAP negative 7833  
## 7 EAP positive 10083  
## 8 EAP sadness 4045  
## 9 EAP surprise 2538  
## 10 EAP trust 5739  
## # ... with 20 more rows

ggplot(count(sentiments, sentiment)) +   
 geom\_col(aes(sentiment, n, fill = sentiment))



ggplot(count(sentiments, author, sentiment)) +   
 geom\_col(aes(sentiment, n, fill = sentiment)) +   
 facet\_wrap(~ author) +  
 coord\_flip() +  
 theme(legend.position = "none")

 Based on afinn, we see the whole text contains more negative words, like score=-5. And Shelly uses more extreme words than others.

Based on bing, we learn three authors are all negative and Shelly has more negative emotions than other two.

Based on nrc, we get the number of words for different emotions for whole text. And then, we can get information for different authors. They pay attention on different emotions.