Section 3:Sentiment Analysis

Step1:word level

1: Using bigrams to provide context in sentiment analysis

Our sentiment analysis approach in simply counted the appearance of positive or negative words, according to a reference lexicon. One of the problems with this approach is that a word's context can matter nearly as much as its presence. For example, the words "happy" and "like" will be counted as positive, even in a sentence like "I'm not happy and I don't like it!"

Now that we have the data organized into bigrams, it's easy to tell how often words are preceded by a word like "not":

```
bigrams_separated %>%
  filter(word1 == "not") %>%
  count(word1, word2, sort = TRUE)
```

```
## # A tibble: 946 x 3
##    word1 word2     n
##    <chr> <chr> <int>
##  1 not   to      139
##  2 not   be      131
##  3 not   the     103
##  4 not   a        88
##  5 not   have     72
##  6 not   only     66
##  7 not   in       57
##  8 not   so       57
##  9 not   even     44
## 10 not   been     37
## # ... with 936 more rows
```

By performing sentiment analysis on the bigram data, we can examine how often sentiment-associated words are preceded by "not" or other negating words. We could use this to ignore or even reverse their contribution to the sentiment score.

Let's use the AFINN lexicon for sentiment analysis, which you may recall gives a numeric sentiment score for each word, with positive or negative numbers indicating the direction of the sentiment.

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

We can then examine the most frequent words that were preceded by "not" and were associated with a sentiment.

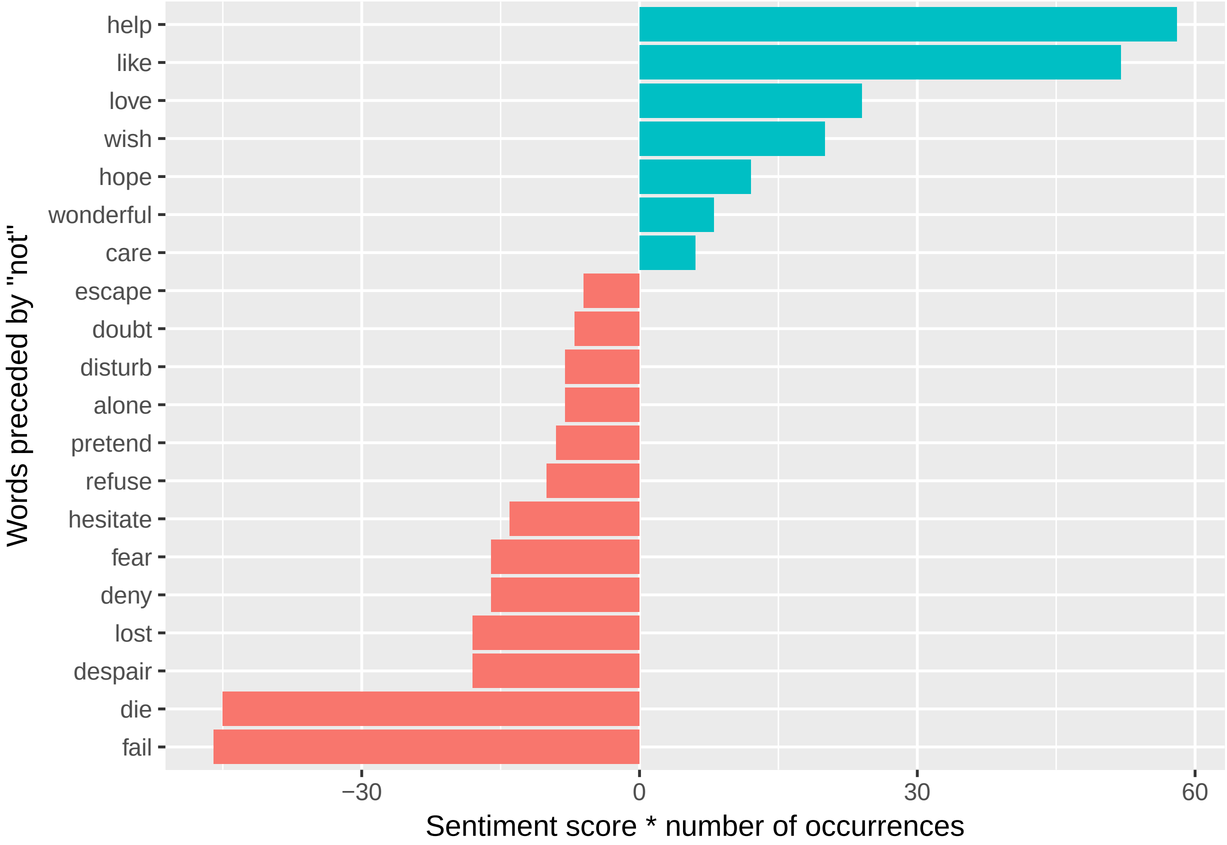
```
not_words<-bigrams_separated %>%
  filter(word1 == "not") %>%
  inner_join(AFINN, by = c(word2 = "word")) %>%
  count(word2, score, sort = TRUE) %>%
  ungroup()
not_words
```

```
## # A tibble: 158 x 3
##    word2   score     n
##    <chr>   <int> <int>
##  1 help        2    29
##  2 like        2    26
##   3 fail       -2    23
##   4 wish        1    20
##   5 die        -3    15
##   6 pretend    -1     9
##   7 deny       -2     8
##   8 fear       -2     8
##   9 love        3     8
##  10 doubt      -1     7
## # ... with 148 more rows
```

For example, the most common sentiment-associated word to follow "not" was "help", which would normally have a (positive) score of 2.

It's worth asking which words contributed the most in the "wrong" direction. To compute that, we can multiply their score by the number of times they appear (so that a word with a score of +3 occurring 10 times has as much impact as a word with a sentiment score of +1 occurring 30 times). We visualize the result with a bar plot.

```
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()
```



The 20 words preceded by ‘not’ that had the greatest contribution to sentiment scores, in either a positive or negative direction. The bigrams “not help” and “not like” were overwhelmingly the largest causes of misidentification, making the text seem much more positive than it is. But we can see phrases like “not fail” and “not die” sometimes suggest text is more negative than it is.