Assignment4-Task 2

Shicong Wang

11/29/2021

Task ONE: Pick a book

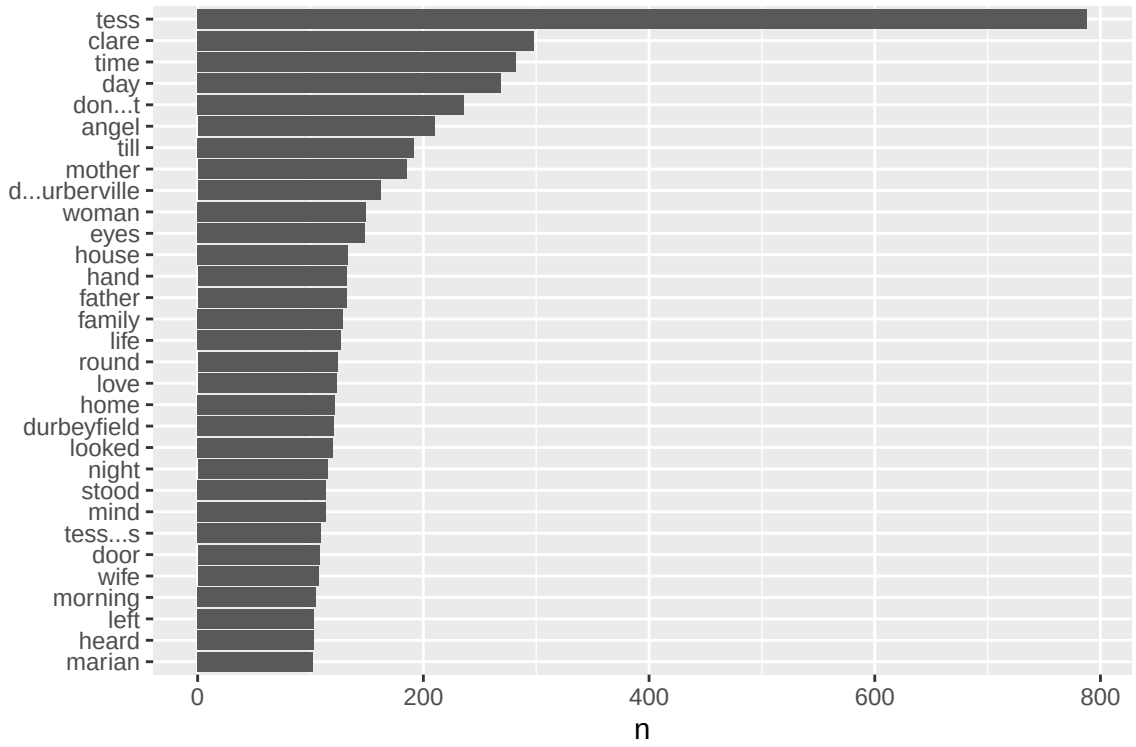
I choose the book <Tess of the d'Urbervilles>, whose author is Thomas Hardy. Here is tje book link: Tess of the d'Urbervilles

Firstly, it's necessary to tidy the book, which means we need to break the text into individual tokens (a process called tokenization) and transform it to a tidy data structure. Also, use mutate() to annotate a line number quantity to keep track of lines in the original format and a chapter (using a regex) to find where all the chapters are. Now that the data is in one-word-per-row format, we can manipulate it with tidy tools like dplyr. Often in text analysis, we will want to remove stop words; stop words are words that are not useful for an analysis, typically extremely common words such as "the", "of", "to", and so forth in English. We can remove stop words (kept in the tidy text data set stop_words) with an anti_join().

Task TWO: bag of words analysis

the frequency of the word

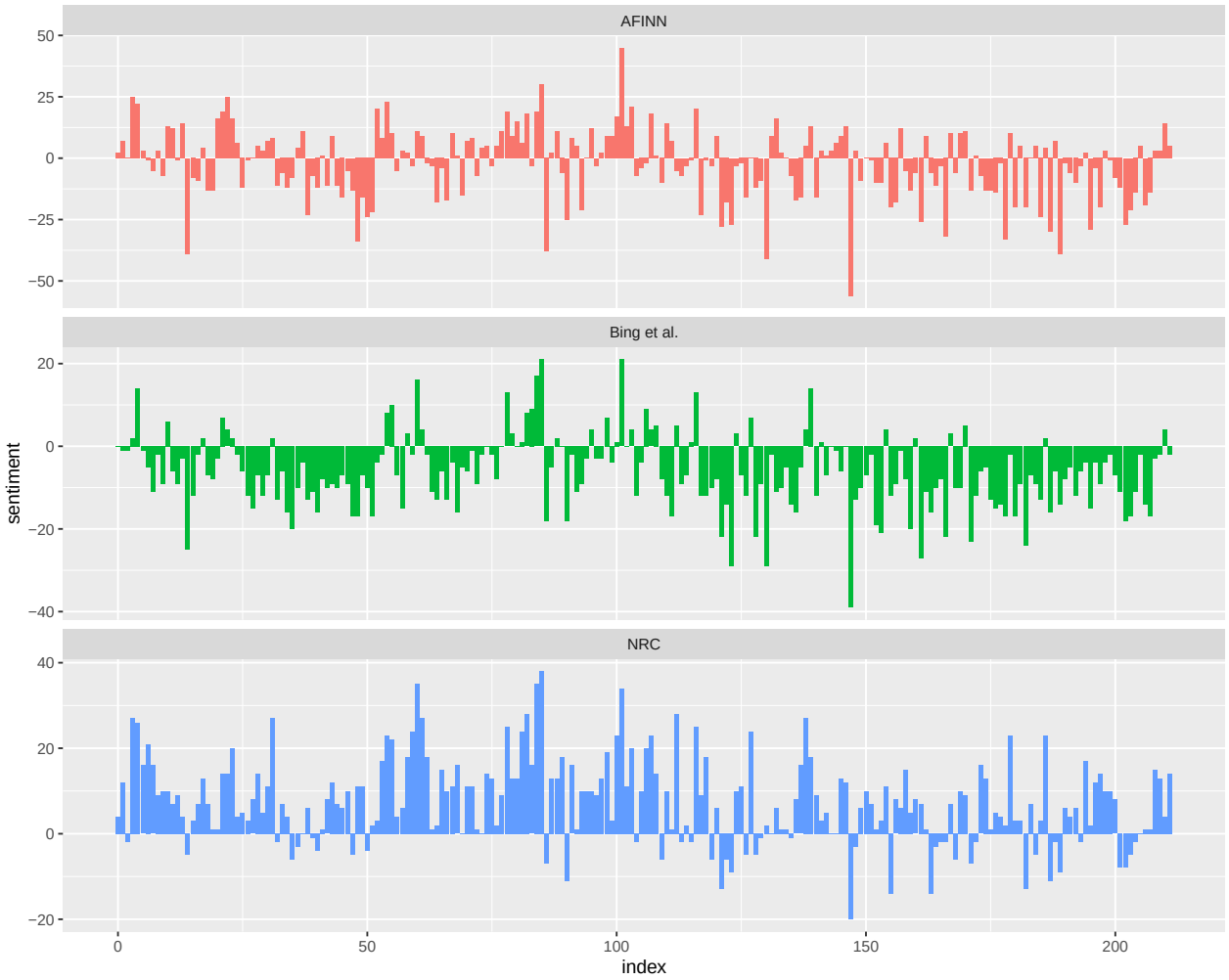
Because we've been using tidy tools, our word counts are stored in a tidy data frame. This allows us to pipe this directly to the ggplot2 package, for example to create a visualization of the most common words.



choose an index length and a sentiment dictionary

Let's address the topic of opinion mining or sentiment analysis. When human readers approach a text, we use our understanding of the emotional intent of words to infer whether a section of text is positive or negative, or perhaps characterized by some other more nuanced emotion like surprise or disgust. We can use the tools of text mining to approach the emotional content of text programmatically.

There are a variety of methods and dictionaries that exist for evaluating the opinion or emotion in text. The tidytext package provides access to several sentiment lexicons. Three general-purpose lexicons are AFINN, bing, and nrc.



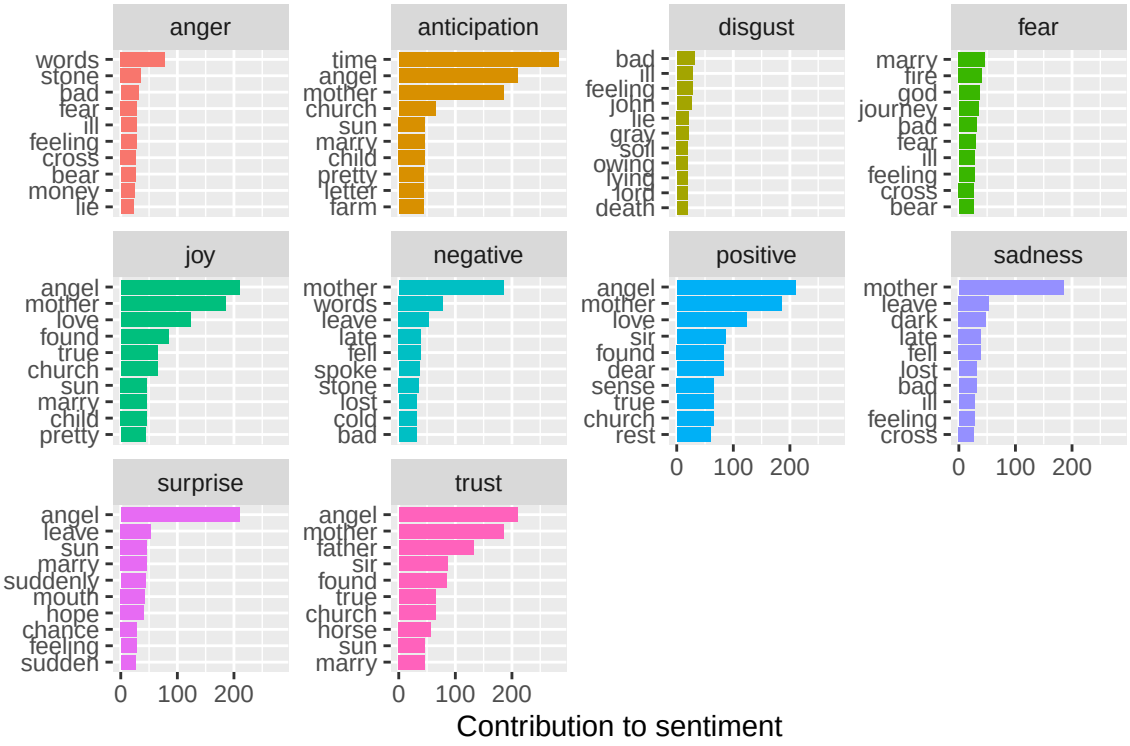
We find similar differences between the methods when looking at other novels; the NRC sentiment is high, the AFINN sentiment has more variance, the Bing et al. sentiment appears to find longer stretches of similar text, but all three agree roughly on the overall trends in the sentiment through a narrative arc.

Most common positive and negative words

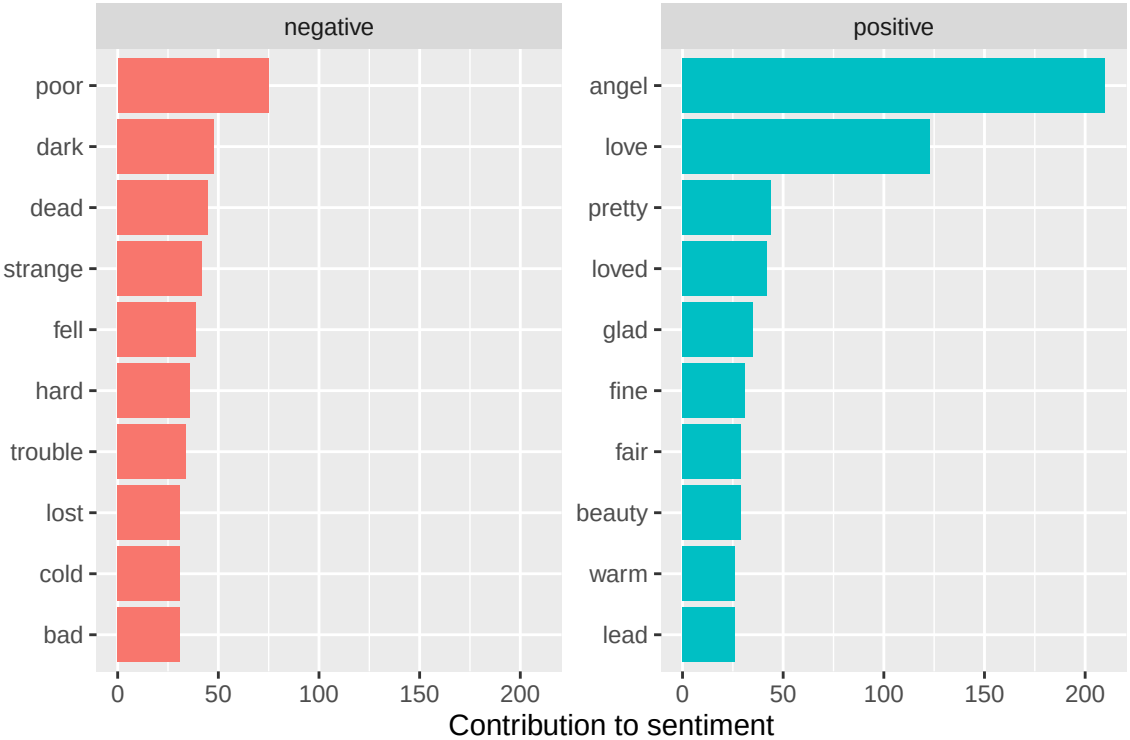
One advantage of having the data frame with both sentiment and word is that we can analyze word counts that contribute to each sentiment. By implementing count() here with arguments of both word and sentiment, we find out how much each word contributed to each sentiment.

This can be shown visually, and we can pipe straight into ggplot2, if we like, because of the way we are consistently using tools built for handling tidy data frames.

lexicon "nrc"



lexicon "bing"



word cloud

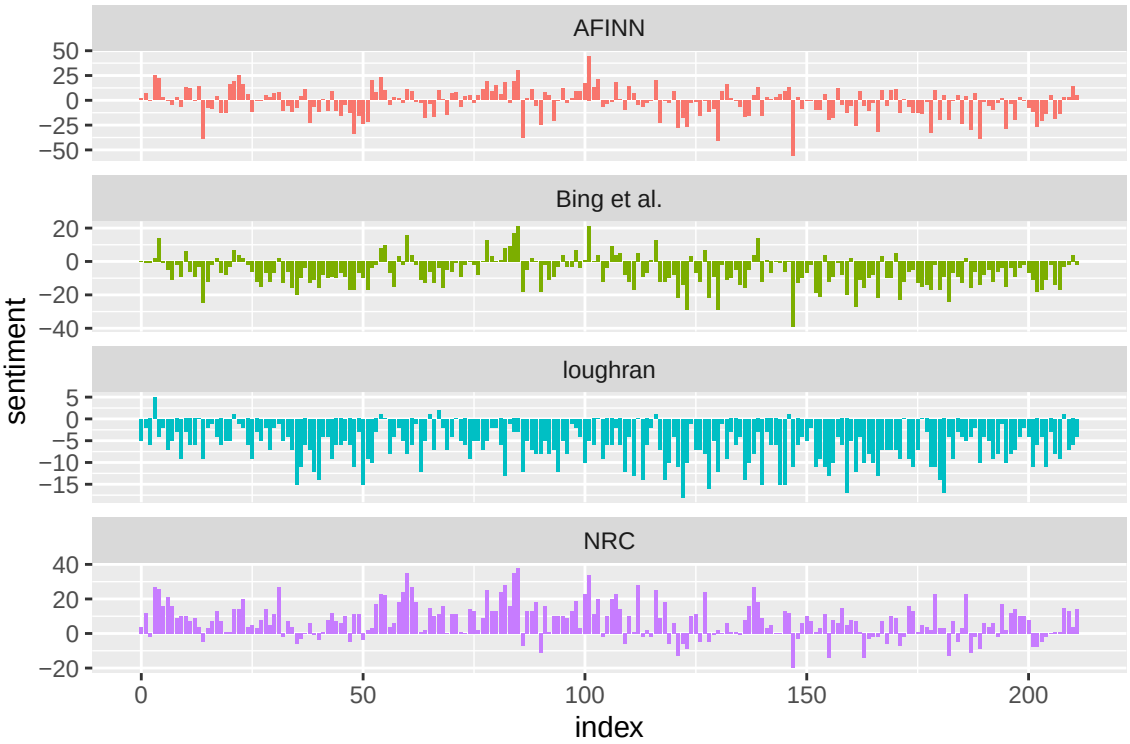
Let's do the sentiment analysis to tag positive and negative words using an inner join, then find the most common positive and negative words. Until the step where we need to send the data to comparison.cloud(), this can all be done with joins, piping, and dplyr because our data is in tidy format.

The size of a word's text is in proportion to its frequency within its sentiment, and the color of words represents different emotions. We can use this visualization to see the most important positive and negative

words.

Extra Credits

In this section, I choose a new lexicon named "loughran". This lexicon divided words into constraining, litigious, negative, positive, superfluous and uncertainty. As we need to compare positive and negative words, select positive and negative parts contained in the lexicon.



Since loughran was developed based on analyses of of financial sentiment terms, and intentionally avoids words like "share" and "fool", as well as subtler terms like "liability" and "risk" that may not have a negative meaning in a financial context. As a result, it may not be that suitable to this novel. However, if we need to make analysis in financial report in future, we can use this sentiment in a proper way.