**How To Do Sentiment Analysis On Your Favourite Book With R**

I love dissecting and analysing my favourite books by reading it again and again, discussing it with my like-minded friends, reading about the authors and other people’s writing about them. My obsession with books lead me to thinking, how can I visualise them in interesting ways? Making Word Cloud is an easy and fun way to visualise your favourite book as in [the previous post](https://www.mydatahack.com/how-to-create-a-word-cloud-for-your-favourite-book-with-r/). In this post, we will go down deeper into the world of text analytics by using [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis). I will show you how to split the text by sentence, conduct sentence-wise sentiment analysis and create an interactive plot that shows how sentiment changes as the story progresses.

There are many different ways to do sentiment analysis. The easiest way is to split the sentence by word and score each word by looking up word sentiment dictionary. [The tidytext packages](https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html) in R has a build in function to do a basic sentiment analysis. The package documentation from CRAN shows sentiment analysis on Jane Austin text. [This one with Harry Potter](http://uc-r.github.io/sentiment_analysis) is also fun to read. For more advanced [Natural Language Processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP), you can use [Stanford CoreNLP](https://stanfordnlp.github.io/CoreNLP/), which is very powerful, but cumbersome to use and can be slow at a time.

In this post, I am using [sentimentr](https://github.com/trinker/sentimentr). It does sentence-wise sentiment analysis by using dictionary lookup and valence shifter. With valence shifter, it updates the normal word sentiment dictionary look up score with the score by looking at polarised word (such as really or hardly). It reflects the sentiment more accurately than just scoring individual words. It also has the get\_sentense function to splice the text into sentences by using regular expressions.

The sentimentr package really makes sentiment analysis pop and accessible. I really love using it.

Sentence analysis on literature is hard because of slangs, unconventional style and expressions, unambiguous sentence separation, sarcasms, jokes and so on. Despite the outcome is not perfect, the package offers us a lot of fun!

As the continuation from [the previous post](https://www.mydatahack.com/how-to-create-a-word-cloud-for-your-favourite-book-with-r/), I am using one of my favourite book, [*On The Road*](https://en.wikipedia.org/wiki/On_the_Road) by the greatest Beat hero [Jack Kerouac](https://en.wikipedia.org/wiki/Jack_Kerouac).

Ok, let’s code!

**Summary Steps**

* Download a pdf file and convert it into text.
* Create a data frame with row per sentence and do clean up.
* Further split the sentence by get\_sentence() and create a new data frame
* Get sentiment per sentence.
* Plot the sentiment by plotly.
* Check the results and have fun!

**Steps**

1. Get the pdf file of On The Road from [freeditorial.com](https://freeditorial.com/) and use [pdftools](https://ropensci.org/blog/2016/03/01/pdftools-and-jeroen/) to convert it to text. This step is the same as in [the previous post](https://www.mydatahack.com/how-to-create-a-word-cloud-for-your-favourite-book-with-r/).

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

1. Create the first data frame with row by sentence. For sentiment analysis, cleaning up the text has to be a little bit more diligent than just making a word cloud. You just need to look at your text file to determine what is needed. The regular expression is the way to go to clean up the text data. Use online resource for [detailed examples](http://biostat.mc.vanderbilt.edu/wiki/pub/Main/SvetlanaEdenRFiles/regExprTalk.pdf) as well as a [quick reference](http://www.endmemo.com/program/R/gsub.php). Make sure to test your regex with a short example!

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

1. Further split the sentences by the power of get\_sentences function. This step is necessary to get a sentiment score per sentence according to the sentimentr package. The sentiment function in sentimentr may returns more than one sentiment score per sentence if you do not do this.

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

1. Once your data frame is ready, do the scoring with sentiment(). Add negative, positive and neutral indicator for visualisation.

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

1. You can plot it with base R plot function. I like using Plotly because it becomes interactive. The x-axis shows the progress of the story and the y-axis shows the sentiment. Positive, negative and neutral sentiments are colour-coded. Once you hover the mouse over the dot, you can see the sentence. The magrittr package enables the magic of the pipe (%>%) operation.

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

1. Let’s check what is the most positive and the most negative and calculate the average sentiment score. I also created csv files with sentiment score above 1 and below -1.

[cc lang="rsplus" tab\_size="4" lines="-1"]

[/cc]

**Discussion**

The average sentiment score was 0.024. All the highs and lows must have cancelled each other out, coming out slightly positive just like life in general.

The most positive moment in the story according to sentimentr is absolutely gold.

*I think Marylou was very, very wise leaving you, Dean, said Galatea*

Yeah, that’s pretty much the best decision! On a more serious note, this is the prime example of the effect of using valence shifter in the sentiment algorithm. It has two positive polarised words (*very, very*) on a positive word (*wise*). Very twice really amplified the positive sentiment on this sentence.

Not sure about the most negative one. But, this is also the fun part, getting misplaced sentiment by an algorithm.

*He showed me rooming houses where he stayed, railroad hotels, poolhalls, diners, sidings where he jumped off the engine for grapes, Chinese restaurants where he ate, park*

Looking at the results, I think it performed well overall despite questionable sores here and there. To be fair, On The Road is a difficult one to run sentiment algorithm because it is written in a stream of conscious style with many unique expressions and unclear sentence break points.

Here are the top 6 postive sentences. No.2 is the classic Dean Moriarty haunted by his own madness. No.4 and 5 are questionable. No.6 is absolutely yes, I think.

|  |
| --- |
| I think Marylou was very, very wise leaving you, Dean, said Galatea |
| Hmm, ah, yes, excellent, splendid, harrumph, egad! |
| It certainly was pleasant, said Hingham, looking away |
| Doll, Dorothy and Roy Johnson, a girl from Buffalo, Wyoming, who was Dorothy's friend, Stan, Tim Gray, Babe, me, Ed Dunkel, Tom Snark, and several others, thirteen in all |
| There's one last thing I want to know - But, dear Sal, you're listening, you're sitting there, we'll ask Sal |
| Absolutely, now, yes? |

Here are the bottom 6 negative sentences. Yeah, it certainly shows some low moments in the book despite of some questionable entries. No. 5 is another classic Dean Moriarty.

|  |
| --- |
| He showed me rooming houses where he stayed, railroad hotels, poolhalls, diners, sidings where he jumped off the engine for grapes, Chinese restaurants where he ate, park benches where he met girls, and certain places where he'd done nothing but sit and wait around |
| There was something paralyzed about his movements, and he did nothing about leaving the doorway, but just stood in it, muttering, Stan, and Don't go, and looking after us anxiously as we rounded the comer |
| Then the third day began having a terrible series of waking nightmares, and the were so absolutely horrible and grisly and green that I lay there doubled up with my hands around my knees, saying, 'Oh, oh, oh, ah, oh |
| Offisah, I heard Dean say in the most unctuous and ridiculous tones, offisah, I was only buttoning my flah |
| And the one to my left here, older, more sure of himself but sad |
| But this foolish gang was bending onward |

Wow, what I fun I had!

Now, you are ready for sentiment analysis for your favourite book!