Name: Abhirath Kapoor Roll Number: 101803047 Group: COE 3

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

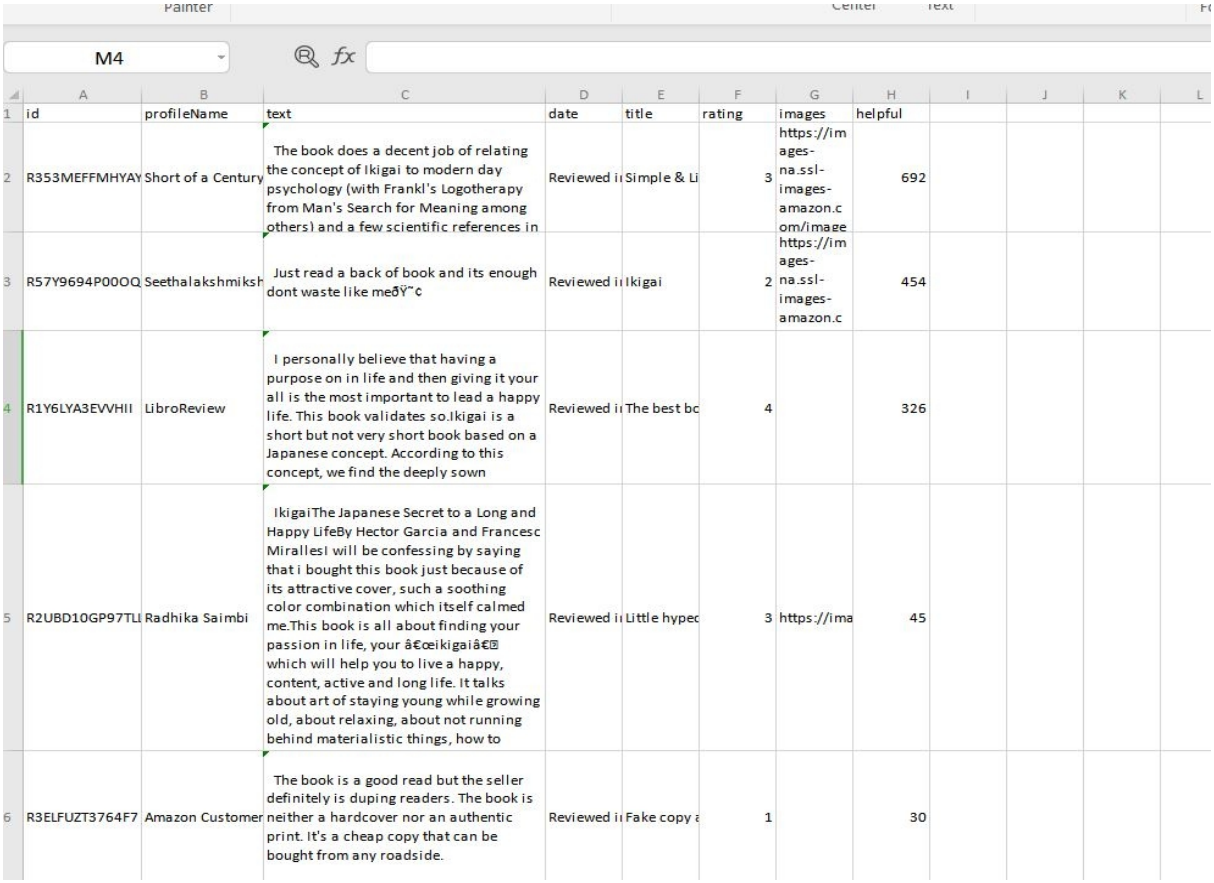
Text Mining and Sentiment Analysis: Analysis with R

R is a language and environment for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques and is highly extensible. R is available as free software. It's easy to learn and use and can produce well designed publication-quality plots.

DataSet is about the reviews given to a product(Book- Ikigai) on Amazon.

The input file for this has six columns but we will be dealing with only one column, the "text" of reviews and is a csv file.

A sample of the first few rows are shown:



	A	B	C	D	E	F	G	H
1	id	profileName	text	date	title	rating	images	helpful
2	R353MEFFMHYAY	Short of a Century	The book does a decent job of relating the concept of Ikigai to modern day psychology (with Frankl's Logotherapy from Man's Search for Meaning among others) and a few scientific references in	Reviewed i	Simple & Li	3	https://images-na.ssl-images-amazon.com/image	692
3	R57Y9694P00OQ	Seethalakshmiksh	Just read a back of book and its enough dont waste like meðŸ˜¢	Reviewed i	Ikigai	2	https://images-na.ssl-images-amazon.c	454
4	R1Y6LYA3EVVHII	LibroReview	I personally believe that having a purpose on in life and then giving it your all is the most important to lead a happy life. This book validates so.Ikigai is a short but not very short book based on a Japanese concept. According to this concept, we find the deeply sown	Reviewed i	The best bc	4		326
5	R2UBD10GP97TLI	Radhika Saimbi	IkigaiThe Japanese Secret to a Long and Happy LifeBy Hector Garcia and Francesc MiralllesI will be confessing by saying that i bought this book just because of its attractive cover, such a soothing color combination which itself calmed me.This book is all about finding your passion in life, your â€œikigaiâ€ which will help you to live a happy, content, active and long life. It talks about art of staying young while growing old, about relaxing, about not running behind materialistic things, how to	Reviewed i	Little hypec	3	https://ima	45
6	R3ELFUZT3764F7	Amazon Customer	The book is a good read but the seller definitely is duping readers. The book is neither a hardcover nor an authentic print. It's a cheap copy that can be bought from any roadside.	Reviewed i	Fake copy a	1		30

Name: Abhirath Kapoor Roll Number: 101803047 Group: COE 3

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

Installing and loading R packages

library(tm) - for text mining operations like removing numbers, special characters, punctuation and stop words (Stop words in any language are the most commonly occurring words that have very little value for NLP and should be filtered out. Examples of stop words in English are “the”, “is”, “are”.)
library(wordcloud) - for generating the word cloud plot.
library(syuzhet) - for sentiment scores and emotion classification.
library(snowballc) - for stemming, which is the process of reducing words to their base or root form.

Reading file data into R

read.csv() is used for reading comma-separated value (csv) files, where a comma “,” is used a field separator

```
4
5  reviews <- read.csv(file.choose(), header = T)
6
7  str(reviews)
8
```

```
> reviews <- read.csv(file.choose(), header = T)
>
> str(reviews)
'data.frame':   1948 obs. of  1 variable:
 $ text: chr  "\n  The book does a decent job of relating the concept of Ikigai to modern day psychology (with Frankl's Logoth"| __truncated__ "\n  Just
 read a back of book and its enough dont waste like meðŸ\230¢\n" "\n  I personally believe that having a purpose on in life and then giving it your all i
s the most important to "| __truncated__ "\n  IkigaiThe Japanese Secret to a Long and Happy LifeBy Hector Garcia and Francesc MirallesI will be confessi
n"| __truncated__ ...
>
```

In your R script, add the following code to load the data into a corpus.

```
8
9  corpus <- iconv(reviews$text)
10 corpus <- Corpus(VectorSource(corpus))
11
12 inspect(corpus[1:5])
```

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

Cleaning up Text Data

Cleaning the text data starts with making transformations like converting the complete text to a lower case as it is case sensitive along with removing special characters from the text. This is done using the `tm_map()` function to replace special characters like `/`, `@` and `|` with a space. The next step is to remove the unnecessary whitespace and convert the text to lower case.

Then remove the *stopwords*. They are the most commonly occurring words in a language and have very little value in terms of gaining useful information. They should be removed before performing further analysis. Examples of stopwords in English are "the, is, at, on". There is no single universal list of stop words used by all NLP tools. `stopwords` in the `tm_map()` function supports several languages like English, French, German, Italian, and Spanish. Please note the language names are case sensitive.

```
13
14  corpus <- tm_map(corpus, tolower)
15
16  corpus <- tm_map(corpus, removePunctuation)
17
18  corpus <- tm_map(corpus, removeNumbers)
19
20  corpus <- tm_map(corpus, removeWords, stopwords("english"))
21
22  corpus <- tm_map(corpus, removeWords, c("book", "read", "life"))
23
24  corpus <- tm_map(corpus, stripWhitespace)
25
26
```

The last step is text stemming. It is the process of reducing the word to its root form. The stemming process simplifies the word to its common origin.

```
corpus <- tm_map(corpus, stemDocument)
inspect(corpus[1:5])
```

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

```
> corpus <- tm_map(corpus, tolower)
Warning message:
In tm_map.SimpleCorpus(corpus, tolower) : transformation drops documents
>
> corpus <- tm_map(corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(corpus, removePunctuation) :
  transformation drops documents
>
> corpus <- tm_map(corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(corpus, removeNumbers) :
  transformation drops documents
>
> corpus <- tm_map(corpus, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")) :
  transformation drops documents
>
> corpus <- tm_map(corpus, removeWords, c("book", "read", "life"))
Warning message:
In tm_map.SimpleCorpus(corpus, removeWords, c("book", "read", "life")) :
  transformation drops documents
>
> corpus <- tm_map(corpus, stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(corpus, stripWhitespace) :
  transformation drops documents
```

Building the term document matrix

After cleaning the text data, the next step is to count the occurrence of each word, to identify popular or trending topics. Using the function `TermDocumentMatrix()` from the text mining package, you can build a Document Matrix – a table containing the frequency of words.

In your R script, add the following code and run it to see the frequency of different words in different reviews.

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

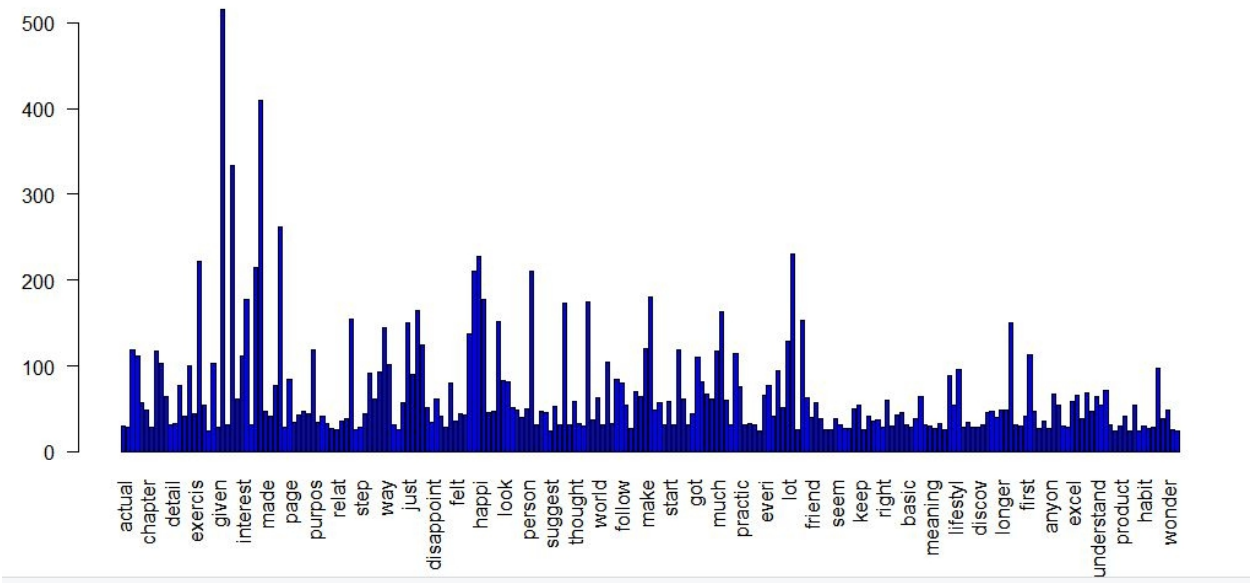
```
29
30  reviews_final <- corpus
31
32  tdm <- TermDocumentMatrix(reviews_final)
33  tdm <- as.matrix(tdm)
34  tdm[1:10, 1:5]
35
```

```
                Docs
Terms            1 2 3 4 5
  actual         1 0 0 0 0
  advic          1 0 0 0 0
  also           4 0 2 0 0
  among          1 0 0 0 0
  applic         1 0 0 0 0
  artificiallycr 1 0 0 0 0
  assuag         1 0 0 0 0
  attent         1 0 0 0 0
  author         2 0 0 0 0
  behind         1 0 0 1 0
> w <- rowSums(tdm)
> w <- subset(w, w>=25)
```

Plotting the words with frequencies more than 25 using a bar chart is a good basic way to visualize this word frequent data. In your R script, add the following code and run it to generate a bar chart, which will display in the *Plots* sections of RStudio.

```
35
36  w <- rowSums(tdm)
37  w <- subset(w, w>=25)
38  barplot(w, las = 2, col = "blue")
39
```

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter



One could interpret the following from this bar chart:

- The most frequently occurring word is “good”. Also notice that negative words like “not” don’t feature in the bar chart, which indicates there are no negative prefixes to change the context or meaning of the word “good” (In short, this indicates most responses don’t mention negative phrases like “not good”).
- “happiness”, “want” and “mind” are the next three most frequently occurring words, which indicate that most people feel good about the book.

Generate the Word Cloud

A word cloud is one of the most popular ways to visualize and analyze qualitative data. It’s an image composed of keywords found within a body of text, where the size of each word indicates its frequency in that body of text. Use the word frequency data frame (table) created previously to generate the word cloud. In your R script, add the following code and run it to generate the word cloud and display it in the *Plots* section of RStudio.

Name: Abhirath Kapoor Roll Number: 101803047 Group: COE 3

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

```
62
63 set.seed(1234)
64 wordcloud(words = names(w) ,freq = w, max.words = 100, random.order = F, min.freq = 5, rot.per=0.40,
65 colors = brewer.pal(8,"Dark2"),scale = c(3,0,.3))
66
```

Below is a brief description of the arguments used in the word cloud function;

- **words** – words to be plotted
- **freq** – frequencies of words
- **min.freq** – words whose frequency is at or above this threshold value is plotted (in this case, I have set it to 5)
- **max.words** – the maximum number of words to display on the plot (in the code above, I have set it 100)
- **random.order** – I have set it to FALSE, so the words are plotted in order of decreasing frequency
- **rot.per** – the percentage of words that are displayed as vertical text (with 90-degree rotation). I have set it 0.40 (40 %), please feel free to adjust this setting to suit your preferences
- **colors** – changes word colors going from lowest to highest frequencies
- **scale** – shows the ratio of scale of biggest and smallest entry
-

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

Emotion Classification

Emotion classification is built on the NRC Word-Emotion Association Lexicon. The definition of NRC is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive).

To understand this, we use the `get_nrc_sentiments` function, which returns a data frame with each row representing a sentence from the original file. The data frame has ten columns (one column for each of the eight emotions, one column for positive sentiment valence and one for negative sentiment valence). The data in the columns (anger, anticipation, disgust, fear, joy, sadness, surprise, trust, negative, positive) can be accessed individually or in sets.

Add the following line to your R script and run it, to see the data frame generated from the previous execution of the `get_nrc_sentiment` function.

```
66
67 sentiment_data <- iconv(reviews$text)
68 s <- get_nrc_sentiment(sentiment_data)
69 str(sentiment_data)
70 s[1:10,]
71
```

```
> s[1:10,]
   anger anticipation disgust fear joy sadness surprise trust negative positive
1      1            7       1    3   9       3        3    11        6       24
2      0            0       1    0   0       0        0     0        1        0
3      3           14       1    2   9       3        1    13        6       22
4      2           10       2    1   9       4        3    12        4       14
5      0            1       0    0   2       0        1     2        2        2
6      0            0       0    0   1       0        0     2        0        2
7      0            0       0    0   0       0        0     1        0        1
8      1            0       0    0   1       1        1     3        2        6
9      0            2       0    0   3       0        0     5        0        3
10     1            3       1    0   2       2        0     4        2       11
>
```

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

To get the main column that we require for sentiment analysis, we subtract negative count from the positive and store it in a new column called score. This score will be used to decide whether the review was positive or not.

```
71
72  s$score <- s$positive - s$negative
73  s[1:10,]
74
```

Now when we have obtained this matrix of sentiments, we store it in a separate csv file.

Also, we created barplot of the sum of columns of all the 10 sentiments along with the score value to decide what the average emotion of the review csv file is and as we can see, the most occurring words belong to the positive sentiment, followed by trust and joy. The words belonging to the negative sentiment are very less thus making less effect to the final score.

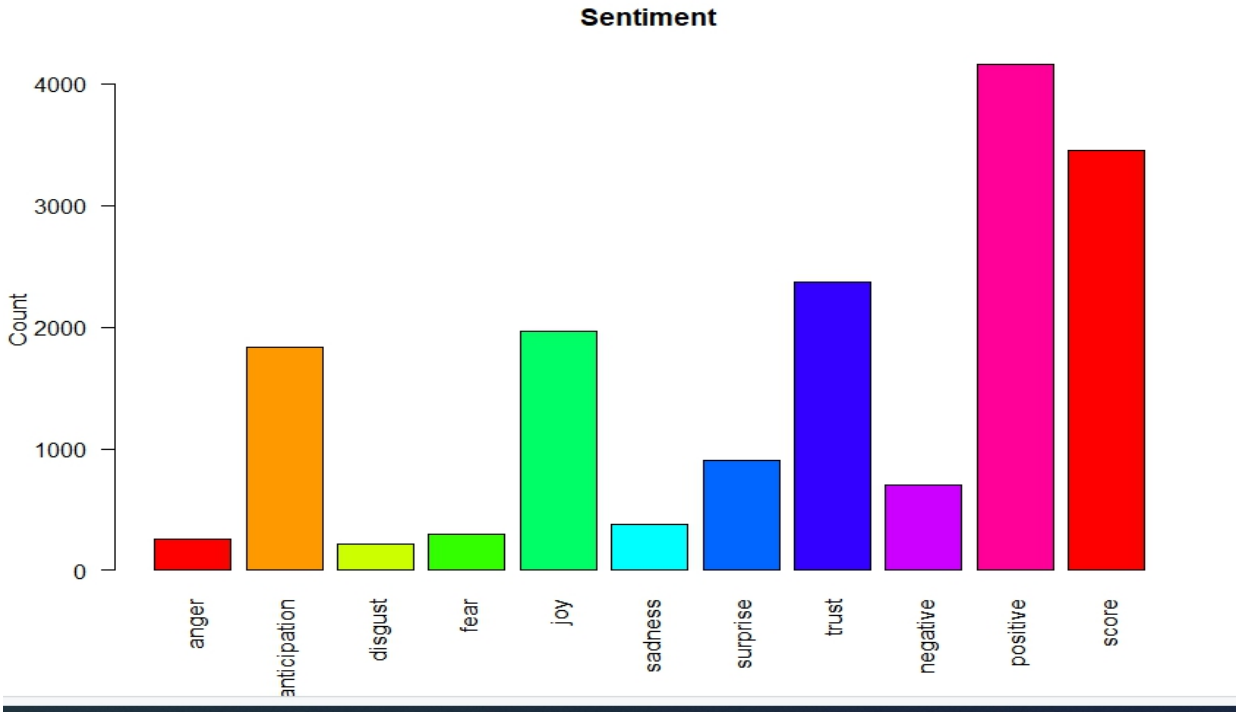
```
write.csv(x = s, file = "C:/Users/Abhirath/Desktop/Final_score.csv")

review_score <- colSums(s[,])
print(review_score)

barplot(colSums(s), las = 2, col = rainbow(10), ylab = 'Count', main = 'Sentiment')
```

Name: Abhirath Kapoor Roll Number: 101803047 Group: COE 3

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter



Below is the Final_score.csv

	A	anger	anticipation	disgust	fear	joy	sadness	surprise	trust	negative	positive	score
2	1	1	7	1	3	9	3	3	11	6	24	18
3	2	0	0	1	0	0	0	0	0	1	0	-1
4	3	3	14	1	2	9	3	1	13	6	22	16
5	4	2	10	2	1	9	4	3	12	4	14	10
6	5	0	1	0	0	2	0	1	2	2	2	0
7	6	0	0	0	0	1	0	0	2	0	2	2
8	7	0	0	0	0	0	0	0	1	0	1	1
9	8	1	0	0	0	1	1	1	3	2	6	4
10	9	0	2	0	0	3	0	0	5	0	3	3
11	10	1	3	1	0	2	2	0	4	2	11	9
12	11	0	2	0	0	3	0	2	4	0	4	4
13	12	1	14	1	4	15	5	2	13	4	27	23
14	13	1	4	0	2	3	2	2	4	2	6	4
15	14	0	3	1	1	3	0	1	2	2	5	3
16	15	1	2	1	0	1	0	1	1	2	3	1
17	16	1	3	1	2	3	2	2	4	6	6	0
18	17	1	3	0	0	3	0	1	4	2	5	3
19	18	1	5	0	1	7	0	1	6	1	20	19
20	19	1	1	0	1	1	2	0	0	3	2	-1
21	20	2	10	2	1	7	3	2	13	7	18	11
22	21	3	14	1	3	14	8	8	15	8	26	18
23	22	1	2	2	1	4	2	2	2	2	8	6
24	23	0	6	0	2	3	0	2	2	2	6	4
25	24	0	0	0	0	0	0	0	0	0	2	2
26	25	0	0	0	0	0	0	0	0	0	1	1
27	26	0	1	0	0	1	0	1	2	0	2	2
28	27	0	2	0	0	6	0	2	5	1	10	9
29	28	0	6	0	1	8	1	3	7	1	15	14
30	29	1	2	0	1	3	1	0	2	3	5	2

Data set Source: Self Extracted Amazon Review Data set
https://www.amazon.in/Ikigai-H%C3%A9ctor-Garc%C3%ADa/dp/178633089X
Reviews of this Book extracted using Amazon reviews exporter

Word Association

This technique can be used effectively to analyze which words occur most often in association with the most frequently occurring words in the survey responses, which helps to see the context around these words.

This script shows which words are most frequently associated with the three terms that are mentioned as parameters (corlimit = 0.19 is the lower limit/threshold).

```
tdm1 <- TermDocumentMatrix(reviews_final)

findAssocs(tdm1, terms = c("good","great","happi"), corlimit=0.19)
```

```
> findAssocs(tdm1, terms = c("good","great","happi"), corlimit=0.19)
$good
keep
0.19

$great
     consider     ikigairel inspirationif          mist     seekersit     territori           tug
          0.2           0.2           0.2           0.2           0.2           0.2           0.2

$happi
                          long                          activ
                          0.44                          0.33
                         smile                          find
                          0.30                          0.28
                         thing                        secret
                          0.26                          0.26
                         alway                          live
                          0.26                          0.25
```