Program – 11

AIM: Write a program to find TF-IDF for any dataset and also plot resultant term frequency matrix.

Introduction and Theory

TF-IDF

A central question in text mining and natural language processing is how to quantify what a document is about. Can we do this by looking at the words that make up the document? One measure of how important a word may be is its term frequency (tf), how frequently a word occurs in a document. There are words in a document, however, that occur many times but may not be important; in English, these are probably words like "the", "is", "of", and so forth. We might take the approach of adding words like these to a list of stop words and removing them before analysis, but it is possible that some of these words might be more important in some documents than others. A list of stop words is not a sophisticated approach to adjusting term frequency for commonly used words.

Another approach is to look at a term's inverse document frequency (idf), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. This can be combined with term frequency to calculate a term's tf-idf, the frequency of a term adjusted for how rarely it is used. It is intended to measure how important a word is to a document in a collection (or corpus) of documents. It is a rule-of-thumb or heuristic quantity; while it has proved useful in text mining, search engines, etc., its theoretical foundations are considered less than firm by information theory experts. The inverse document frequency for any given term is defined as

$$idf(term) = \ln\left(\frac{n_{documents}}{n_{documents\ containing\ term}}\right)$$

Then finally the resulting TF-IDF matrix is then calculated as:

$$tfidf(term, document, Dataset) = tf(term, doc) \times ifd(term, Dataset)$$

The resulting matrix is not normalized, this is done using the L2 normalization:

$$\hat{v} = \frac{\vec{v}}{||\vec{v}||}$$

Program – 11

Code

TF-IDF.R

```
1  library(tm)
2  library(proxy)
3  library(dplyr)
4  library(ggplot2)
5  library(corrplot)
6  print("The dataset: each sentence is one docoment")
7  doc <- c( "The sky is blue.", "The sun is bright today.", "The sun in the
   sky is bright.", "We can see the shining sun, the bright sun.", "The moon
   is full, the sky full of stars.", "The sky was dark, the stars plentiful
   and bright.", "The sun is but a morning star.")
8  corpus <- Corpus( VectorSource(doc) )
9  controlList <- list(removePunctuation = TRUE, stopwords = TRUE, tolower =
   TRUE)
10 print("computing the term-frequency matrix: ")
11 (tf <- as.matrix(TermDocumentMatrix(corpus, control = controlList) ) )
12 corrplot(tf, method = "number", is.corr = FALSE, cl.pos = "n")
13 print("computing the idf, and then converting into a diagonal matrix (used
   later)")
14 (idf <- log(ncol(tf) / (1 + rowSums(tf != 0) ) ) )
15 (idf <- diag(idf) )
16 print("calculating the final tf-idf matrix")
17 tf_idf <- crossprod(tf, idf)
18 colnames(tf_idf) <- rownames(tf)
19 (tf_idf <- tf_idf / sqrt(rowSums(tf_idf^2) ) )
20 corrplot(tf_idf, method = "number")
```

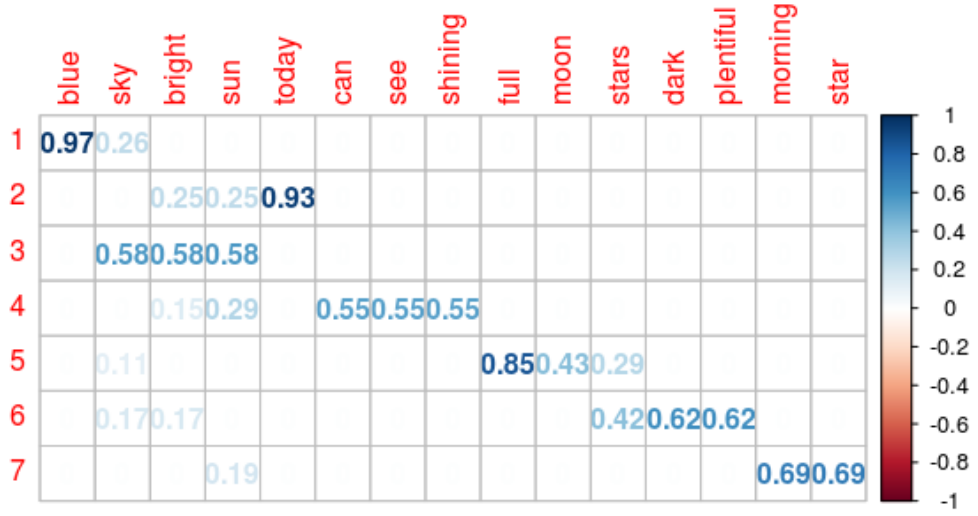
Results & Outputs

```
~/Documents/R programs/
> library(tm)
> library(proxy)
> library(dplyr)
> library(ggplot2)
> library(corrplot)
>
> print("The dataset: each sentence is one docoment")
[1] "The dataset: each sentence is one docoment"
> doc <- c( "The sky is blue.", "The sun is bright today.", "The sun in the sky is bright.", "We can see the shining sun, the bright sun.", "The moon is full, the
 sky full of stars.", "The sky was dark, the stars plentiful and bright.", "The sun is but a morning star.")
> corpus <- Corpus( VectorSource(doc) )
> controlList <- list(removePunctuation = TRUE, stopwords = TRUE, tolower = TRUE)
>
> print("computing the term-frequency matrix: ")
[1] "computing the term-frequency matrix: "
> (tf <- as.matrix(TermDocumentMatrix(corpus, control = controlList) ) )
           Docs
Terms       1 2 3 4 5 6 7
  blue      1 0 0 0 0 0 0
  sky       1 0 1 0 1 1 0
  bright    0 1 1 1 0 1 0
  sun       0 1 1 2 0 0 1
  today     0 1 0 0 0 0 0
  can       0 0 0 1 0 0 0
  see       0 0 0 1 0 0 0
  shining   0 0 0 1 0 0 0
  full      0 0 0 0 2 0 0
  moon      0 0 0 0 1 0 0
  stars     0 0 0 0 1 1 0
  dark      0 0 0 0 0 1 0
  plentiful 0 0 0 0 0 1 0
  morning   0 0 0 0 0 0 1
  star      0 0 0 0 0 0 1
> corrplot(tf, method = "number", is.corr = FALSE, cl.pos = "n")
>
> print("computing the idf, and then converting into a diagonal matrix (used later)")
[1] "computing the idf, and then converting into a diagonal matrix (used later)"
> (idf <- log(ncol(tf) / (1 + rowSums(tf != 0) ) ) )
     blue       sky    bright       sun     today       can       see   shining      full      moon     stars      dark plentiful   morning      star
1.2527630 0.3364722 0.3364722 0.3364722 1.2527630 1.2527630 1.2527630 1.2527630 1.2527630 1.2527630 0.8472979 1.2527630 1.2527630 1.2527630 1.2527630
```

Program – 11

```
~/Documents/R programs/
  plentiful 0 0 0 0 0 1 0
  morning   0 0 0 0 0 0 1
  star      0 0 0 0 0 0 1
> corrplot(tf, method = "number", is.corr = FALSE, cl.pos = "n")
>
> print("computing the idf, and then converting into a diagonal matrix (used later)")
[1] "computing the idf, and then converting into a diagonal matrix (used later)"
> (idf <- log(ncol(tf) / (1 + rowSums(tf != 0)) ) )
     blue       sky    bright       sun     today       can       see   shining      full      moon     stars      dark plentiful   morning      star
1.2527630 0.3364722 0.3364722 0.3364722 1.2527630 1.2527630 1.2527630 1.2527630 1.2527630 1.2527630 0.8472979 1.2527630 1.2527630 1.2527630 1.2527630
> (idf <- diag(idf) )
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]    [,15]
 [1,] 1.252763 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [2,] 0.000000 0.3364722 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [3,] 0.000000 0.0000000 0.3364722 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [4,] 0.000000 0.0000000 0.0000000 0.3364722 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [5,] 0.000000 0.0000000 0.0000000 0.0000000 1.252763 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [6,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 1.252763 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [7,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 1.252763 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [8,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 1.252763 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
 [9,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 1.252763 0.000000 0.0000000 0.000000 0.000000 0.000000 0.000000
[10,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.252763 0.0000000 0.000000 0.000000 0.000000 0.000000
[11,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.8472979 0.000000 0.000000 0.000000 0.000000
[12,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 1.252763 0.000000 0.000000 0.000000
[13,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 1.252763 0.000000 0.000000
[14,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 1.252763 0.000000
[15,] 0.000000 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.000000 0.000000 0.000000 1.252763
>
> print("calculating the final tf-idf matrix")
[1] "calculating the final tf-idf matrix"
> tf_idf <- crossprod(tf, idf)
> colnames(tf_idf) <- rownames(tf)
> (tf_idf <- tf_idf / sqrt(rowSums(tf_idf^2) ) )

Docs      blue       sky    bright       sun     today       can       see   shining      full      moon     stars      dark plentiful   morning      star
   1 0.9657724 0.2593911 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
   2 0.0000000 0.0000000 0.2510818 0.2510818 0.9348347 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
   3 0.0000000 0.5773503 0.5773503 0.5773503 0.0000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
   4 0.0000000 0.0000000 0.1465097 0.2930193 0.0000000 0.545489 0.545489 0.545489 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
   5 0.0000000 0.1142179 0.0000000 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.850519 0.4252595 0.2876214 0.0000000 0.0000000 0.0000000 0.0000000
   6 0.0000000 0.1665139 0.1665139 0.0000000 0.0000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.4193120 0.6199692 0.6199692 0.0000000 0.0000000
   7 0.0000000 0.0000000 0.0000000 0.0000000 0.1865826 0.0000000 0.000000 0.000000 0.000000 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.6946895 0.6946895
> corrplot(tf_idf, method = "number")
```



Program – 11

Findings and Learnings:

1. TF-IDF if one of the most popular methods in text processing
2. R provides easy to use tools for performing text analysis.
3. We have successfully implemented TF-IDF in R.