Assignment No. 6

Aim: To compute TF-IDF (Term Frequency-Inverse Document Frequency) values of words from different types of corpora using R programming. The analysis will include:

1. A corpus with unique values.
2. A corpus with similar documents.
3. A single word repeated multiple times in multiple documents.

Theory:

TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It is commonly used in information retrieval and text mining. TF-IDF is the product of two statistics, term frequency (TF) and inverse document frequency (IDF).

- **Term Frequency (TF):** Measures how frequently a term appears in a document.

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

- **Inverse Document Frequency (IDF):** Measures how important a term is within the entire corpus

$$\text{IDF}(t, D) = \log\left(\frac{\text{Total number of documents in the corpus}}{\text{Number of documents containing term } t}\right)$$

- **TF-IDF:** Combines both measures.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Experiment Details

Implementation in R:

1. **Load the necessary libraries:**

   library(tm)

   library(tidytext)

   library(dplyr)

2. **Create the corpora:**

```
# Corpus with unique values
corpus_unique <- Corpus(VectorSource(c("apple banana cherry", "dog
elephant fish", "grape hat ink")))
```

```
corpus_unique          List of  3
 $ 1:List of 2
  ..$ content: chr "apple banana cherry"
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:04:19"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "1"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ 2:List of 2
  ..$ content: chr "dog elephant fish"
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:04:19"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "2"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ 3:List of 2
  ..$ content: chr "grape hat ink"
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:04:19"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "3"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 - attr(*, "class")= chr [1:2] "SimpleCorpus" "Corpus"
```

```
# Corpus with similar documents
corpus_similar <- Corpus(VectorSource(c("apple apple banana", "apple
banana cherry", "banana cherry apple")))
```

```
corpus_similar         List of  3
 $ 1:List of 2
  ..$ content: chr "apple apple banana"
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:07:03"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "1"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
```

```
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
$ 2:List of 2
 ..$ content: chr "apple banana cherry"
 ..$ meta   :List of 7
 .. ..$ author       : chr(0)
 .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:07:03"
 .. ..$ description  : chr(0)
 .. ..$ heading      : chr(0)
 .. ..$ id           : chr "2"
 .. ..$ language     : chr "en"
 .. ..$ origin       : chr(0)
 .. ..- attr(*, "class")= chr "TextDocumentMeta"
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
```

```
$ 3:List of 2
 ..$ content: chr "banana cherry apple"
 ..$ meta   :List of 7
 .. ..$ author       : chr(0)
 .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:07:03"
 .. ..$ description  : chr(0)
 .. ..$ heading      : chr(0)
 .. ..$ id           : chr "3"
 .. ..$ language     : chr "en"
 .. ..$ origin       : chr(0)
 .. ..- attr(*, "class")= chr "TextDocumentMeta"
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
- attr(*, "class")= chr [1:2] "SimpleCorpus" "Corpus"
```

```
# Corpus with a single word repeated multiple times
corpus_repeated <- Corpus(VectorSource(c("apple apple apple", "apple apple apple", "apple apple apple")))
```

```
corpus_repeated          List of  3
$ 1:List of 2
 ..$ content: chr "apple apple apple"
 ..$ meta   :List of 7
 .. ..$ author       : chr(0)
 .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:08:42"
 .. ..$ description  : chr(0)
 .. ..$ heading      : chr(0)
 .. ..$ id           : chr "1"
 .. ..$ language     : chr "en"
 .. ..$ origin       : chr(0)
 .. ..- attr(*, "class")= chr "TextDocumentMeta"
```

```
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
$ 2:List of 2
 ..$ content: chr "apple apple apple"
 ..$ meta   :List of 7
 .. ..$ author       : chr(0)
 .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:08:42"
 .. ..$ description  : chr(0)
 .. ..$ heading      : chr(0)
 .. ..$ id           : chr "2"
 .. ..$ language     : chr "en"
 .. ..$ origin       : chr(0)
 .. ..- attr(*, "class")= chr "TextDocumentMeta"
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
```

```
$ 3:List of 2
 ..$ content: chr "apple apple apple"
 ..$ meta   :List of 7
 .. ..$ author       : chr(0)
 .. ..$ datetimestamp: POSIXlt[1:1], format: "2024-10-14 19:08:42"
 .. ..$ description  : chr(0)
 .. ..$ heading      : chr(0)
 .. ..$ id           : chr "3"
 .. ..$ language     : chr "en"
 .. ..$ origin       : chr(0)
 .. ..- attr(*, "class")= chr "TextDocumentMeta"
 ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
- attr(*, "class")= chr [1:2] "SimpleCorpus" "Corpus"
```

3. **Preprocess the text:**
   preprocess_corpus <- function(corpus) { corpus <- tm_map(corpus, content_transformer(tolower)) corpus <- tm_map(corpus, removePunctuation) corpus <- tm_map(corpus, removeNumbers) corpus <- tm_map(corpus, removeWords, stopwords("english")) corpus <- tm_map(corpus, stripWhitespace) return(corpus) } corpus_unique <- preprocess_corpus(corpus_unique) corpus_similar <- preprocess_corpus(corpus_similar) corpus_repeated <- preprocess_corpus(corpus_repeated)

Functions	
preprocess_corpus	function (corpus)

4. **Create Document-Term Matrices and compute TF-IDF values:**

   dtm_unique <- DocumentTermMatrix(corpus_unique)
   dtm_similar <- DocumentTermMatrix(corpus_similar)
   dtm_repeated <- DocumentTermMatrix(corpus_repeated)

   tfidf_unique <- weightTfIdf(dtm_unique)
   tfidf_similar <- weightTfIdf(dtm_similar)
   tfidf_repeated <- weightTfIdf(dtm_repeated)

```
dtm_unique              List of  6
    $ i       : int [1:9] 1 1 1 2 2 2 3 3 3
    $ j       : int [1:9] 1 2 3 4 5 6 7 8 9
    $ v       : num [1:9] 1 1 1 1 1 1 1 1 1
    $ nrow    : int 3
    $ ncol    : int 9
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr [1:9] "apple" "banana" "cherry" "dog" ...
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
model                   List of  13
```

```
dtm_similar              List of  6
    $ i        : int [1:8] 1 1 2 2 2 3 3 3
    $ j        : int [1:8] 1 2 1 2 3 1 2 3
    $ v        : num [1:8] 2 1 1 1 1 1 1 1
    $ nrow     : int 3
    $ ncol     : int 3
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr [1:3] "apple" "banana" "cherry"
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
dtm_repeated             List of  6
    $ i        : int [1:3] 1 2 3
    $ j        : int [1:3] 1 1 1
    $ v        : num [1:3] 3 3 3
    $ nrow     : int 3
    $ ncol     : int 1
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr "apple"
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
tfidf_unique             List of  6
    $ i        : int [1:9] 1 1 1 2 2 2 3 3 3
    $ j        : int [1:9] 1 2 3 4 5 6 7 8 9
    $ v        : Named num [1:9] 0.528 0.528 0.528 0.528 0.528 ...
     ..- attr(*, "names")= chr [1:9] "1" "1" "1" "2" ...
    $ nrow     : int 3
    $ ncol     : int 9
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr [1:9] "apple" "banana" "cherry" "dog" ...
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency - inverse document frequency (normalize...
```

```
tfidf_similar            List of  6
    $ i        : int [1:2] 2 3
    $ j        : int [1:2] 3 3
    $ v        : Named num [1:2] 0.195 0.195
     ..- attr(*, "names")= chr [1:2] "2" "3"
    $ nrow     : int 3
    $ ncol     : int 3
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr [1:3] "apple" "banana" "cherry"
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency - inverse document frequency (normalize...
```

```
tfidf_repeated           List of  6
    $ i        : int(0)
    $ j        : int(0)
    $ v        : Named num(0)
     ..- attr(*, "names")= chr(0)
    $ nrow     : int 3
    $ ncol     : int 1
    $ dimnames:List of 2
     ..$ Docs : chr [1:3] "1" "2" "3"
     ..$ Terms: chr "apple"
    - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_triplet_matrix"
    - attr(*, "weighting")= chr [1:2] "term frequency - inverse document frequency (normalize...
```

5. **Convert to data frame for better readability:**

```
tfidf_to_df <- function(tfidf) {
  as.data.frame(as.matrix(tfidf))
}
```

```
df_tfidf_unique <- tfidf_to_df(tfidf_unique)
df_tfidf_similar <- tfidf_to_df(tfidf_similar)
df_tfidf_repeated <- tfidf_to_df(tfidf_repeated)

df_tfidf_unique
df_tfidf_similar
df_tfidf_repeated
```

```
df_tfidf_unique          3 obs. of 9 variables
 $ apple   : num  0.528 0 0
 $ banana  : num  0.528 0 0
 $ cherry  : num  0.528 0 0
 $ dog     : num  0 0.528 0
 $ elephant: num  0 0.528 0
 $ fish    : num  0 0.528 0
 $ grape   : num  0 0 0.528
 $ hat     : num  0 0 0.528
 $ ink     : num  0 0 0.528
```

```
df_tfidf_similar         3 obs. of 3 variables
 $ apple : num  0 0 0
 $ banana: num  0 0 0
 $ cherry: num  0 0.195 0.195
```

```
df_tfidf_repeated        3 obs. of 1 variable
 $ apple: num  0 0 0
```

Conclusion:

In this experiment, we successfully computed TF-IDF values for words from three different types of corpora using R programming:

1. **Corpus with unique values:** Each document had distinct words, leading to a uniform distribution of TF-IDF values.
2. **Corpus with similar documents:** Similar documents resulted in higher TF-IDF values for common words, emphasizing their importance within the corpus.
3. **Single word repeated multiple times:** The repeated word had a high term frequency but a lower inverse document frequency, leading to high TF values but lower TF-IDF values.

The TF-IDF metric effectively highlighted the importance of words relative to the corpus, showcasing its utility in various text mining applications. Further analysis could involve visualizing these TF-IDF values to gain deeper insights.