NAME:VEDASHREE BHALERAO

ROLL NO:2213191

BATCH:B

SUBJECT:BDA LAB ASSIGNMENT NO 6

Big Data Analytics Experiment no. 06

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 with similar documents
```

```
corpus_similar <- Corpus(VectorSource(c("apple apple banana", "apple
banana cherry", "banana cherry apple")))

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

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)

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)

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

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.

CODE:

```
# Install necessary R packages

%%R

install.packages('tm', repos='https://cran.rstudio.com/')

install.packages('tidytext', repos='https://cran.rstudio.com/')

install.packages('dplyr', repos='https://cran.rstudio.com/')


%%R

# Load necessary libraries

library(tm)

library(tidytext)

library(dplyr)


# Step 1: Create the corpora


# Corpus with unique values

corpus_unique <- Corpus(VectorSource(c("apple banana cherry",

                                       "dog elephant fish",

                                       "grape hat ink")))


# Corpus with similar documents

corpus_similar <- Corpus(VectorSource(c("apple apple banana",

                                        "apple banana cherry",

                                        "banana cherry apple")))
```

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

# Step 2: Preprocess the text
preprocess_corpus <- function(corpus) {
  corpus <- tm_map(corpus, content_transformer(tolower))     # Convert to lower case
  corpus <- tm_map(corpus, removePunctuation)               # Remove punctuation
  corpus <- tm_map(corpus, removeNumbers)                   # Remove numbers
  corpus <- tm_map(corpus, removeWords, stopwords("english")) # Remove stopwords
  corpus <- tm_map(corpus, stripWhitespace)                 # Strip whitespace
  return(corpus)
}

# Apply preprocessing
corpus_unique <- preprocess_corpus(corpus_unique)
corpus_similar <- preprocess_corpus(corpus_similar)
corpus_repeated <- preprocess_corpus(corpus_repeated)

# Step 3: Create Document-Term Matrices and compute TF-IDF values

# Document-Term Matrices
dtm_unique <- DocumentTermMatrix(corpus_unique)
dtm_similar <- DocumentTermMatrix(corpus_similar)
dtm_repeated <- DocumentTermMatrix(corpus_repeated)

# Compute TF-IDF
```

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

# Step 4: Convert to data frame for better readability
tfidf_to_df <- function(tfidf) {
  return(as.data.frame(as.matrix(tfidf)))
}

# Convert TF-IDF matrices to data frames
df_tfidf_unique <- tfidf_to_df(tfidf_unique)
df_tfidf_similar <- tfidf_to_df(tfidf_similar)
df_tfidf_repeated <- tfidf_to_df(tfidf_repeated)

# Step 5: Display the TF-IDF values
cat("TF-IDF for Unique Corpus:\n")
print(df_tfidf_unique)

cat("\nTF-IDF for Similar Corpus:\n")
print(df_tfidf_similar)

cat("\nTF-IDF for Repeated Word Corpus:\n")
print(df_tfidf_repeated)
```

OUTPUT:

```
TF-IDF for Unique Corpus:
      apple    banana    cherry       dog  elephant      fish     grape
1 0.5283208 0.5283208 0.5283208 0.0000000 0.0000000 0.0000000 0.0000000
2 0.0000000 0.0000000 0.0000000 0.5283208 0.5283208 0.5283208 0.0000000
3 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.5283208
        hat       ink
1 0.0000000 0.0000000
2 0.0000000 0.0000000
3 0.5283208 0.5283208

TF-IDF for Similar Corpus:
  apple banana    cherry
1     0      0 0.0000000
2     0      0 0.1949875
3     0      0 0.1949875

TF-IDF for Repeated Word Corpus:
  apple
1     0
2     0
3     0
```