MediaSentiment Package

Mackenzie Ramalho

2024-04-01

The MediaSentiment Package will provide a set of functions to assist in sentiment analysis tasks. This document serves as a guide to understand these functions effectively.

Function 1: ‘clean_text()’

This function cleans text data by removing special characters, numbers, and extra white spaces.

```r
# Function to clean text
clean_text <- function(text) {
  # Remove special characters
  cleaned_text <- gsub("[^a-zA-Z\\s]", "", text)
  # Remove extra white spaces
  cleaned_text <- gsub("\\s+", " ", cleaned_text)
  # Convert to lowercase
  cleaned_text <- tolower(cleaned_text)
  return(cleaned_text)
}

#Example
cleaned_text <- clean_text("Sample text with special characters: !@#$%^&*()")

cleaned_text
```

```
## [1] "sampletextwithspecialcharacters"
```

Function 2: ‘tokenize_text()’

This function tokenizes the text into individual words or tokens.

```r
# Function to tokenize text
tokenize_text <- function(text) {
  tokens <- strsplit(text, "\\s+")[[1]]
  return(tokens)
}

#Example
tokens <- tokenize_text("This is a sample sentence.")

tokens
```

```
## [1] "This"      "is"        "a"         "sample"    "sentence."
```

Function 3: 'calculate_word_frequency()'

This calculates the frequency of each word in the text.

```
# Function to calculate word frequency
calculate_word_frequency <- function(text) {
  tokens <- tokenize_text(text)
  word_freq <- table(tokens)
  return(word_freq)
}

#Example
word_freq <- calculate_word_frequency("This is a sample sentence. This sentence is a sample.")

word_freq
```

```
## tokens
##          a         is     sample    sample.   sentence  sentence.       This
##          2          2          1          1          1          1          2
```

Function 4: 'find_keywords()'

This function identifies keywords in the text based on their frequency or relevance.

```
# Function to find keywords
find_keywords <- function(text, threshold = 1) {
  word_freq <- calculate_word_frequency(text)
  keywords <- names(word_freq[word_freq >= threshold])
  return(keywords)
}

#Example
keywords <- find_keywords("This is a sample sentence. Sample sentence contains sample words.")

keywords
```

```
## [1] "a"         "contains"  "is"        "sample"    "Sample"    "sentence"
## [7] "sentence." "This"      "words."
```

Function 5: 'sentiment_analysis()'

This performs sentiment analsis on text.

```
# Function for sentiment analysis
sentiment_analysis <- function(text) {
  # Will replace words with more media focused phrases.
  positive_words <- c("good", "positive", "happy")
  negative_words <- c("bad", "negative", "sad")
  tokens <- tokenize_text(text)

  positive_count <- sum(tokens %in% positive_words)
  negative_count <- sum(tokens %in% negative_words)

  if (positive_count > negative_count) {
    return("Positive")
  } else if (negative_count > positive_count) {
    return("Negative")
  } else {
    return("Neutral")
  }
}

#Example
sentiment <- sentiment_analysis("This is a positive sentence.")

sentiment
```

```
## [1] "Positive"
```