Lenguajes de programación - T08: Implementación de la distancia Damerau-Levenshtein en el lenguaje R

Jorge Aurelio Morales Manrique
C.C. 1010075711
jomorales@unal.edu.co

Universidad Nacional de Colombia
Abril 13 de 2021

5. Manual de usuario

En la carpeta adjunta con el nombre "Código Fuente", se encuentra un proyecto realizado en RStudio, entorno de desarrollo integrado para el lenguaje de programación R, dedicado a la computación estadística y gráficos. A continuación se lista la serie de pasos necesarios para ejecutar el programa (omitir los pasos 1 y 2 en caso de tener R y RStudio instalados en su sistema).

1. Descargar e instalar R de la página oficial https://www.r-project.org/
2. Descargar e instalar RStudio de la página oficial https://www.rstudio.com/products/rstudio/download/
3. Descargar la carpeta **damerou_levenshtein**
4. Abrir RStudio e importar el proyecto descargado en el paso anterior.
5. Abrir el archivo **main.R**, debe aparecer en la sección de trabajo.
6. En el archivo **data.txt** ingresar las palabras del lenguaje una por línea sin espacios al final de cada una. Añadir un salto de línea al final del archivo. A continuación se muestra la estructura antes mencionada.

```
1| if          7| double        13| function
2| else        8| float         14| true
3| do          9| string        15| false
4| while      10| char          16| assert
5| for        11| void          17| break
6| int        12| null          18| (EMPTY)
```

7. En el panel donde fue abierto el archivo **main.R** realizar la siguiente combinación de teclas **(Ctrl + A)** para seleccionar todo el código, posteriormente dar click en el botón **Run** ubicado en la parte superior del panel mencionado anteriormente. Esto realizará la ejecución del programa.

8. En la consola se podrán observar los valores de la distancia promedio y la varianza. Abrir el explorador de archivos y dirigirise a la carpeta **images** en la cual estará la gráfica del histograma generado en formato **PNG**. En la carpeta **results** podrá encontrar un archivo de texto con el nombre **distances.txt** el cual contiene la matriz triangular superior con las distancias generadas para cada par de palabras en el lenguaje.

6. Manual técnico

La estructura interna del programa se divide en tres componentes principales los cuales se describen a continuación

Función distancia D-L

Toma como parámetros dos palabras a, b y calcula su distancia D-L.

```r
dldistance <- function(a, b) {
  n1 <- nchar(a)
  n2 <- nchar(b)
  d = matrix(0, nrow = n1 + 1, ncol = n2 + 1)
  for (index in 1:dim(d)[1]) {
    d[index, 1] <- index - 1
  }
  for (index in 1:dim(d)[2]) {
    d[1, index] <- index - 1
  }
  for (i in 2:dim(d)[1]) {
    for (j in 2:dim(d)[2]) {
      cost <- 0
      if (substring(a, i - 1, i - 1) != substring(b, j - 1, j - 1)) {
        cost <- 1
      }
      options <- c(d[i - 1, j] + 1, d[i, j - 1] + 1, d[i - 1, j - 1] + cost)
      d[i, j] <- min(options)
      if (i > 2 && j > 2 && substring(a, i - 1, i - 1) == substring(b, j - 2, j - 2)
      && substring(a, i - 2, i - 2) == substring(b, j - 1, j - 1)) {
        options <- c(d[i, j], d[i - 2, j - 2] + 1)
        d[i, j] <- min(options)
      }
    }
  }
  return(d[n1 + 1, n2 + 1])
}
```

Figura 1: Función que calcula la distancia D-L entre dos palabras.

Función distancias del lenguaje

Lee las palabras del lenguaje desde el archivo de texto **data.txt**, crea una matriz triangular superior de tamaño $(n + 1)(n + 1)$ y calcula la distancia entre cada par de palabras leídas previamente, haciendo uso de la primera función. Finalmente guarda la matriz mencionada anteriormente en un archivo de texto y la retorna como resultado.

```r
language_distances <- function() {
  # plot histogram
  # mean and variance
  words <- scan('data.txt', what='', sep='\n')
  n <- length(words)
  distances <- matrix(nrow = n + 1, ncol = n + 1)
  distances[1, 1] = '/'
  for (i in 2:(n+1)) {
    distances[1, i] <- words[i - 1]
  }
  for (i in 2:(n+1)) {
    distances[i, 1] <- words[i - 1]
  }
  for (i in 2:(n+1)) {
    for (j in 2:(n+1)) {
      if (j > i) {
        distances[i, j] <- dldistance(words[i - 1], words[j - 1])
      }
    }
  }
  write.table(distances, file="results/distances.txt",
  row.names=FALSE, col.names=FALSE, sep = '\t\t')
  return(distances)
}
```

Figura 2: Cálculo de las distancias entre las palabras del lenguaje.

Función principal

Genera el histograma de las distancias generadas en la función anterior y lo guarda en un archivo con formato **PNG**. Finalmente calcula la distancia promedio y la varianza e imprime dichos valores en consola.

```r
main <- function() {
  distances <- language_distances()
  n = dim(distances)[1] - 1
  elements <- array(0L, dim = c(n * (n-1) / 2))
  index = 1
  for (i in 2:(n+1)) {
    for (j in 2:(n+1)) {
      if (j > i) {
        value <- strtoi(distances[i, j], base = 0L)
        elements[index] <- value
        index <- index + 1
      }
    }
  }
  png('images/histogram.png', width = 600, height = 600)
  hist(elements, xlab = 'Distance')
  print(c('Distancia promedio: ', mean(elements)))
  print(c('Varianza', var(elements)))
  dev.off()
}
```

Figura 3: Función principal.