Supuesto práctico 1- SSOO - 2023-2024
Analizando datos

Introducción

La comunicación entre el usuario y los sistemas LINUX/UNIX se realiza mediante el intérprete de comando o Shell, además de ser un lenguaje de programación muy potente y flexible. La realización de *scripts* para facilitar tareas de administración, gestión, instalación, control, monitorización, búsquedas, procesado de información, y un largo etcétera, es habitual y proporciona un gran rendimiento.

Objetivo

Desarrollar una aplicación que analice el texto proveniente de un conjunto de correos electrónicos y trate de aportar información sobre si el correo electrónico es potencialmente peligroso y puede ser clasificado como **spam**. Para llevar a cabo este proceso se va a disponer de 2 ficheros, **Fraud_word.txt** y **Emails.txt** [1]. **Fraud_word.txt** contiene un amplio conjunto de palabras y expresiones que son ampliamente utilizadas en correos spam por ciberdelincuentes. **Emails.txt** contiene el texto de casi 12000 correos electrónicos, entre los que hay correos peligrosos que tratan de hacerse con datos sensibles de los destinatarios, correos masivos y correos normales.

Se trata de desarrollar una aplicación que en base al conjunto de palabras que se proporcionan, compute de forma independiente cuántas veces aparece cada una en todos y cada uno de los correos electrónicos proporcionados. Posteriormente, en base a los resultados alcanzados se podrá predecir si un correo electrónico podría ser potencialmente peligroso o no.
La aplicación a desarrollar proporcionará un menú, desde el cual se accederá a todas las funcionalidades.

---

1. Análisis de datos
2. Predicción
3. Informes de resultados
4. Ayuda
5. Salir

La opción 1, **Análisis de datos** se encargará:

1. Pedir al usuario el nombre del fichero en el que se encuentran las palabras o términos a buscar, el nombre del fichero donde se almacenan los correos electrónicos y el nombre del fichero donde se desea almacenar el resultado del análisis. Comprobar que los datos introducidos son correctos, los ficheros de palabras y correos electrónicos existen y que no existe ya un fichero con el mismo nombre para guardar el resultado.

2. El análisis se realizará sobre una matriz de N x M filas, donde cada fila "N" se corresponderá con un correo electrónico y cada columna "M" con una expresión. Cada fila de la matriz almacenará los datos correspondientes al fichero **Emails.txt**, una columna para el identificador y la columna de la etiqueta (donde un 0 es ham y un 1 es spam. A continuación, tantas columnas como términos haya en fichero **Fraud_words.txt**.

Ejemplo

Id.	spam/ham	"exp1"	"exp2"	…..	"expM"
1	0				
…….					
N	1				

1. Datos provenientes del dataset: https://www.kaggle.com/datasets/llabhishekll/fraud-email-dataset/code?select=fraud_email_.csv

Supuesto práctico 1- SSOO - 2023-2024
Analizando datos

Antes de realizar el análisis se eliminarán todos los signos de puntuación del texto a analizar y a continuación, se procederá a calcular el número de veces que cada expresión, de forma individual, aparece en cada correo electrónico. Es decir, calculará la frecuencia con que cada expresión se encuentra presente en cada uno de los correos electrónicos.

Para evitar problemas con mayúsculas, minúsculas, símbolos especiales y dígitos, se eliminarán los símbolos especiales y dígitos y tanto expresiones como correos electrónicos se tratarán en minúsculas. Los resultados se irán almacenando en la matriz propuesta y al finalizar el análisis, éste se guardará en el fichero cuyo nombre será el indicado durante la petición de datos al usuario y tendrá extensión "**.freq".**
El anexo 2 os proporciona algunos comandos básicos de awk que os ayudarán durante el análisis.

La opción 2, en primer lugar, deberá preguntar si se acaba de realizar el análisis o si, por el contrario, se desea cargar el análisis de frecuencias desde un fichero. Si necesita cargar la matriz, pedirá el fichero de frecuencias y con sus datos cargará la matriz para posteriormente realizar el siguiente cálculo. Si también existirea el fichero con extensión **".tfidf"**, se informaría al usuario que el fichero con la métrica TF_idf también se ha realizado con anterioridad y se pide conformidad para cargar todos los datos en las matrices correspondientes.

Si sólo existen los datos de frecuencia, tras cargar la matriz correspondiente, a continuación, se procederá a realizar su análisis para calcular la métrica TF_idf, que vamos a utilizar para predecir, en base a la frecuencia de cada una de las expresiones, si un correo es spam o ham. Si una instancia tiene un TF-IDF alto vamos a predecir que ese correo podría ser spam; en caso contrario diremos que es ham.

El cálculo de esta métrica se realizará sobre una nueva matriz con las mismas filas y columnas que la anterior, a la que se añade una columna que nos servirá para predecir si tras el análisis el correo es considerado spam o ham. El cálculo tiene como fuente de información la matriz donde se ha almacenado la frecuencia de cada término para cada correo electrónico. El resultado de este análisis se guardará con el mismo nombre que el fichero con extensión "**.freq"**, pero tendrá extensión "**.tfidf"**.

La explicación del cálculo de TF-IDF se encuentra en el anexo I.

La opción 3 permite realizar diferentes informes, siempre utilizando un formato tabla con sus correspondientes encabezados:

1. Informe en formato fila/columna donde por cada término muestre en cuantos correos electrónicos del conjunto de datos analizado aparece.
2. Informe donde para un término particular, solicitado al usuario, se muestren los correos electrónicos donde aparece. Del correo electrónico sólo se mostrarán los 50 primeros caracteres.
3. Dado un identificador de correo electrónico, mostrar cuantos términos de los analizados aparecen.

La opción 4 (Ayuda) mostrará ayuda relativa a la aplicación.
La opción 5 (Salir) finaliza la aplicación.

El control de errores es fundamental en cualquier programa y define su calidad, evitando así la insatisfacción del usuario. En el programa se deberán de controlar todos los casos que puedan dar lugar a error.

Fecha de Entrega: 27 de Octubre de 2023

Supuesto práctico 1- SSOO - 2023-2024
Analizando datos

Anexo I

TF es la abreviatura de "Term Frequency" y es el número de veces que un término aparece en un documento (correo electrónico en este caso), comparado con el número de palabras del documento (correo electrónico en este caso).
IDF es la abreviatura de "Inverse Document Frequency" que define la proporción de documentos (correos electrónicos) en el corpus que contienen dicho término.

$$TF = \frac{\text{nº de veces que aparece un término en un documento}}{\text{nº total de términos en el documento}}$$

$$IDF = log(\frac{\text{nº de documentos en el corpus}}{\text{nº documentos en el corpus que contienen el término}})$$

El TF-IDF de un término se calcula multiplicando TF e IDF: **TF-IDF = TF x IDF.** Testa métrica muestra la importancia de un término en un documento del corpus y es muy útil en aplicaciones de procesamiento de lenguaje natural.

Ejemplo:

Tenemos un término o expresión que aparece 10 veces en un correo electrónico compuesto por un total de 100 palabras. Term Frequency (TF) se calcula:

$$TF = \frac{10}{100} = 0{,}1$$

Supongamos que el número total de correos electrónicos analizados son 10000 y 100 de estos correos electrónicos contienen dicho término o expresión. Inverse Document Frequency (IDF) se calcula:

$$IDF = \log_{10} \frac{10000}{100} = 2$$

Una vez calculados ambos valores, TF-IDF = $TF - IDF = 0{,}1 \times 2 = 0{,}2$.

Un valor de tf-idf alto se da cuando hay una frecuencia alta en el documento y baja en el conjunto completo de documentos. Calificaremos como spam un correo donde el TF-idf medio sea superior a 0,3.

Supuesto práctico 1- SSOO - 2023-2024
Analizando datos

Anexo 2

Ejemplos que me facilitan la tarea solicitada:

1. Pasar texto a minúsculas: **echo "Hello dear friend, How ARE YOU? " | awk '{print tolower($0)}'**

2. Buscar cadena dentro de otra: **echo "Hello dear friend, How ARE YOU? " | awk '/dear friend/ {print $0}'**

3. Quitar espacios en blanco**: cat Fraud_word.txt|wc -l |awk '{ gsub(/ /,""); print }'**

4. Utilizar una variable con awk, en este caso además para devolviendo sólo las líneas conde se encuentra dicha palabra**:**

```
$ fichero="Emails.txt"
$ palabra=”Dear friend”
$ awk -v var="$palabra" 'index($0,var)' "$fichero"
```

5. Eliminar símbolos especiales:

```
$ cadena="!! Hello dear! do we go out ? at 22:23 is fine for you?"
$ cadena_nueva="${string//[^[:alnum:]]/" "}"
$ echo "Cadena nueva es : $cadena_nueva
```