Proyectos de Lenguajes

Nombre de la aplicación: Estadísticas y evaluación de contenido usando TF-idf

Lenguaje de programacion: Python y C

Módulos del sistema

- 1 Modulo de ingreso
- 2 Modulo de estadísticas
- 3 Modulo de evaluación de contenido

Modulo de ingreso

Input_1: Repositorio de textos en sus maquinas (yo les entrego una memoria con la información que quiero que lea el programa el día de la prueba). El modulo debe de leer los archivos de la memoria y extraerlos y copiarlos a un directorio. Pueden ser 100, 1000, 10000. Los textos a utilizarse serán textos en español.

Cada texto podrá tener extensiones txt o .csv, tener muchas hojas o unas cuantas líneas.

Input _2; palabras que quiero eliminar. Son palabras adicionales que quiera eliminar, es algo que el usuario puede querer o no hacer, pero debe tener la opción.

Input_3: Ingreso de palabras de búsqueda (para el modulo de evaluación de contenido)

Pre-actividades

a) Construir, buscar stopwords

Stopwords, deben de buscar/crear/manipular una base de stopwords. Las stopwords son palabras muy comunes dentro de un texto y que no aportan con mucho por ejemplo: de, a, para, quien, cuando, como, hacia, a, los, las, etc. en español

b) Identificar como hacer stemming de las palabras: es decir dejarle la raíz y eliminar lo adicional por ejemplo, la función de quitar la terminación de la palabra debo poderla activar o desactivar. Ejemplo del stemming.

La palabra cantar (verbo) tiene una raíz, cant, las terminaciones pueden ser canté, canto, cantamos, Deben identificar las palabras y buscar la raiz, y dejar la raíz. La raíz será una palabra a ser contabilizada.

c) Pre-tratamiento de la información.

A cada documento será necesario eliminarle:

- Espacios en blancos,
- Hacer todas las letras minúsculas
- tokenizar el contenido (en este caso particular convertirlo en palabras)
- eliminar stopwords,
- eliminar dígitos
- Cada documento deberá tener asociada las palabras y las mediciones respectivas (tf, idf,)
- Pueden ingresar otras características que les interesen por ejemplo lematizacion. Pero lo básico está arriba

Módulo de estadísticas

El modulo de estadísticas presentará numero de términos por documento, numero de documentos en el Corpus, y los términos con mayor frecuencia.

Términos	Documento 1	Documento 2	……	Documento n
Cantar	1			4
Hijo	1	2	3	0
Conocer	1		1	0

Esta información debe de ser guardada en una estructura o una matriz.

Mostrar las 50 palabras más comunes del Corpus.

El modulo estadístico debe ser complementamente gráfico.

Puede existir la posibilidad que después de ver las estadísticas quiera eliminar x palabra, entonces deberíamos poder ver, modificar e ingresar el input 2.

Modulo de evaluación de contenido

Algunos conceptos primero antes de entrarle a la descripción del módulo:

Tf: frecuencia de un término dentro de un documento d.

Inverse document frequency (Idf). Es una medida de cuanta información esa palabra provee dentro de un Corpus de documentos, es decir si el término es raro o común en todos los documentos.

Tf-idf es una medida de la importancia de la palabra dentro del corpus. Se incrementa proporcionalmente al número de veces que una palabra aparece en el documento, pero si es una palabra común para todos los documentos quiere decir que es general y se convierte en una palabra sin importancia.

Las formulas son:

$$Tf\text{-}idf(t, d, D) = tf(t,d) \times idf(t,D)$$

$$Idf(t, D) = \log(N/|\{d \in D: t \in d\}|$$

En este módulo se requiere que a los documentos que ya se los había limpiado ahora se les aplique el stemming .

El usuario ingresa un texto en el input 3 y la aplicación calculara el tf e idf de sus términos en documento y dentro del corpus. Se generara la matriz de términos vs documentos. Nótese que los términos en el módulo anterior son diferentes, la matriz es diferente debido a que los términos ahora solo son raíz por lo tanto deberán generar una nueva matriz.

Con tf se presentara una grafica de los 10 documentos con mayor número de términos.

Con el idf se presentara en una gráfica los 10 términos más altos en idf en el corpus.

Y se presentaran los términos con tf-idf más altos.

Output: el sistema devolver los archivos que tengan el tf-idf mas altos con respecto al input 3.

Por ejemplo ingreso 2 palabras: palabra 1 y palabra 2. Qué documento o documentos poseen las dos palabras con alta frecuencia y con tf-idf alto? Qué documentos contienen esas palabras.

Observaciones: La entrada dos, asume que ustedes están creando filtros, no eliminando información, los archivos no son para sobrescribirlos o modificarlos. Únicamente filtran información

Los tres módulos deben interactuar entre ellos, y debe ser una sola aplicación integrada, No se aceptaran módulos por separado.

No se puede usar ningún tipo de base de datos. Únicamente las estructuras de los programas Python y C.

Ejemplo:

Documento 1

tf

Termino	Conteo de terminos (tf)
this	1
is	1
a	2
sample	1

Documento 2

Termino	Conteo de terminos (tf)

this	1
is	1
another	2
example	3

La function de tf-idf es =

Tf-idf(t, d, D)= tf(t,d) x idf(t,D)

Idf(t, D) = log(N/|{d ∈ D: t ∈ d}|

N= número de documentos = 2

t(minúscula) = termino

d(minúscula) = numero de documento que contiene el termino t

D(mayúscula) = corpus de documentos

Idf (this, D) = log (2/2) = 0

Idf(example) = log (2/1)= 0.3010

Tf-idf (this, D) = 0

Tf-idf(example) = 3 * 0.3010 = 0.903

El nltk contiene varios paquetes que sirven para esta actividad.

Sugiero Regexptokenizer, itertools, os, nltk, iplot

Entregables:

Los estudiantes deberán realizar un reporte

- Diario
- Código en archivo
- Aplicación corriendo

Fecha de entrega 18 de diciembre de 3 a 6 pm