**INSTITUTO TECNOLÓGICO DE COSTA RICA I SEMESTRE 2020**

**INGENIERÍA EN COMPUTACIÓN**

**IC-8020 RECUPERACIÓN DE INFORMACIÓN TEXTUAL**

**TAREA PROGRAMADA 3**

El proyecto consite en reducir el vocabulario de una colección de artículos de noticias, para hacer factible la aplicación de algoritmos de clasificación supervisada. No se pide aplicar ningún algoritmo de clasificación, solo se pide escoger los términos que serán usados.

La colección usada es un fragmento de la colección ***Reuters21578*** la cual contiene artículos noticiosos. Se usará de base el archivo reut2-001.sgm que contiene 1000 artículos y se que se describe a continuación.

|  |  |
| --- | --- |
| reut2-001.sgm | Archivo con artículos noticiosos. En formato SGML. Ese es un formato más antiguo y general que XML, pero que para nuestros efectos funciona igual. |

A continuación se muestra un ejemplo de los artículos almacenados. Se destacan en negrita la información que va a ser de interés.

|  |
| --- |
| <**REUTERS** **TOPICS="YES"** LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="16536" **NEWID="1216"**>  <DATE> 3-MAR-1987 13:26:49.70</DATE>  **<TOPICS><D>sugar</D></TOPICS>**  <PLACES><D>uk</D><D>colombia</D></PLACES>  <PEOPLE></PEOPLE>  <ORGS></ORGS>  <EXCHANGES></EXCHANGES>  <COMPANIES></COMPANIES>  <UNKNOWN>  &#5;&#5;&#5;C T  &#22;&#22;&#1;f0325&#31;reute  u f BC-COLOMBIA-TO-SELL-SUGA 03-03 0051</UNKNOWN>  <TEXT>&#2;  <TITLE>COLOMBIA TO SELL SUGAR, LONDON TRADERS SAY</TITLE>  <DATELINE> LONDON, March 3 - </DATELINE>**<BODY>Colombia is holding a snap selling tender**  **tonight for one cargo of world market raw sugar, traders said.**  **The sugar is for March 15/April 15 shipment and bids are**  **being sought based on the New York May delivery futures**  **contract, they added.**  **Reuter**  **&#3;</BODY>**</TEXT>  </REUTERS> |

A partir de esa colección se deben generar los siguientes archivos:

|  |  |
| --- | --- |
| clases.txt | Lista de las clases en que se clasifican los artículos. La clase se encuentra en el elemento <TOPICS> del documento.  Este es un archivo de texto, que en cada línea contiene el nombre de la clase y el número de documentos asociados a esa clase. Solo interesan clases que tengan ***minNc*** documentos o más. Donde ***minNc*** es un parámetro que se obtiene al correr el sistema. |
| docs.txt | Para cada artículo (documento), contiene el identificador extraído del atributo NEWID del elemento <REUTERS> y la clase asociada a dicho documento. Solo deben incluir aquellos artículos de la colección que tengan el atributo TOPICS="YES", y que además contengan al menos una clase en <TOPICS>. Si hay más de una clase en ese campo, tomar la primera. Solo se deben tomar en cuenta los documentos de las clases que fueron escogidas en ***clases.txt***. |
| dicc.txt | Vocabulario de la colección. Para cada término que aparece en el campo <BODY> lista ese término y su valor ni. Solo se toman en cuenta términos que aparecen en los documentos escogidos en ***docs.txt***. |

La colección está en inglés. Los términos se extraen del elemento <BODY>. Se toman como términos las secuencia de caracteres que cumplen con las siguientes reglas:

* palabras (secuencia de uno o más caracteres [a-z])
* uno más dígitos con comas opcionales intercaladas (para miles, millones, etc.) seguidos opcionalmente de un punto seguido de uno o más dígitos; en el diccionario se deben omitir las comas:  
   7,030,000 🡺7030000  
   12345 🡺 12345  
   20.7 🡺20.7  
   1,086.11 🡺1086.11
* descartar los números y letras que aparecen en fragmentos de texto como "**&#3;**"

Adicionalmente, se deben eliminar los stopwords que se listan al final de este documento.

==============De aquí para arriba ya todo está hecho :3==========

**Selección de términos**

Como primer criterio de reducción del vocabulario, se deben eliminar todos los términos que aparecen en menos de una cantidad dada de documentos, ***minNi***. Por ejemplo, si minNi=6, descartar los términos que aparecen en 5 o menos documentos. El valor de ***minNi*** es un parámetro que se obtiene al correr el sistema:

Aquí lo que hay que hacer es agarrar el diccionario que es {término, cantidad} que está en la función **crearDicc** y eliminar los que tengan menos del minNi

Como segundo criterio, se debe usar la métrica de ***ganancia de información*** para determinar los mejores ***numMejores*** términos. El valor de ***numMejores*** es un parámetro que se obtiene al correr el sistema. Se debe además determinar para cada documento cuántos de esos ***numMejores*** mejores términos se incluyen en dicho documento:

Aquí se tiene que usar la fórmula que viene abajo y cuando ya está la ganancia de todos los términos calculada entonces hay dos opciones: hacer un txt nuevo donde venga el NEWID del documento y poner cuantos de los numMejores tiene o agarrar el txt que ya existe y agregarle la cantidad de numMejores para sobreescribirlo.

Recuerde que si se tienen clases C = {C1, C2, …, CL} para un término i, se puede usar la siguiente tabla para calcular la ganancia de información de ese término:

|  |  |  |  |
| --- | --- | --- | --- |
|  | termi | ¬termi |  |
| C1 | ni1 | nc1 - ni1 | nc1 |
| C2 | ni2 | nc2 - ni1 | nc2 |
| … | … | … | … |
| CL-1 | ni,L-1 | ncL-1 - niL-1 | ncL-1 |
| CL | niL | ncL - niL | ncL |
|  | ni | N - ni | N |

Usar las fórmulas:

Hay un diccionario que tiene las clases y el número de documentos por clase, ese se crea en la función **clases**. Entonces aquí es la sumatoria de (num de docs de una clase/num docs totales)\*log(num de docs de una clase/num docs totales) de cada clase del diccionario que ya está hecho.

Dónde,

***E(C)***: Entropía de la colección con clases C.

***E(C,termi)***: Entropía de que se obtiene al separar la colección con clases C usando el término i.

***GI(C,termi)***: ganancia de información que se obtiene al usar el término i.

***nik***: número de documentos de clase k que contienen el término i

***nck***: número de documentos en clase k (En la función clases se crea un diccionario que tiene la clase y el número de documentos que tiene esa clase)

***ni***: número de documentos que contienen término i (El diccionario que se crea en la función **crearDicc** tiene el término y la cantidad de docs que tienen ese término)

***N***: número total de documentos de la colección (En la función **dividirArticulos** hay una variable que se llama **contador**, esa tiene el número total de documentos que se procesan)

Se debe entregar:

* Un programa que reciba los siguientes parámetros:
  + ruta del archivo que contiene los artículos noticiosos
  + cantidad de mejores términos que serán escogidos (***numMejores***)
  + cantidad mínima de documentos por clase (***minNc***)
  + cantidad mínima de documentos por término (***minNi***)
  + un prefijo para usar en los nombres de los archivos generados
* Por ejemplo:  
   > selecterm reut2-001.sgm 200 10 6 pr1\_

Cuando ya se terminan los cálculos hay que hacer que lea estas varas, ahorita todo está alambrado para más rápido jaja  
Indicaría que se debe procesar el archivo **reut2-001.sgm** para extraer los mejores **200** términos, solo interesan las clases que tengan **10** o más documentos, y los términos que aparecen en **6** o más documentos; los archivos generados (que se describen a continuación) deben tener el prefijo "**pr1\_**" en sus nombres.

* El programa debe generar y guardar en disco los archivos ***clases.txt***, ***docs.tx***t y ***dicc.txt*** descritos al inicio de este documento. Dichos archivos deben estar en formato de texto; cada dato (clase, doc, término) en líneas aparte y con datos separados por tabs. **NO debe usar JSON como formato de almacenamiento**. Se debe usar el prefijo pasado como parámetro para identificar fácilmente los archivos generados en una corrida. Por ejemplo: pr1\_clases.txt, pr1\_docs.txt y pr1\_dicc.txt.
* El programa debe calcular la entropía total de la colección y la ganancia de información para cada uno de los términos incluidos en ***dicc.txt***.
* El programa debe producir un archivo con una lista con todos los términos en orden alfabético. Para cada término se debe incluir E(C,termi) y GI(C,termi). Al inicio de lista se debe incluir el valor de la entropía total. Llamar al archivo ***gi.txt*** (incluir el prefijo).
* El programa debe producir un archivo con una lista con los mejores términos de acuerdo con su ganancia de información, en orden decreciente. Para cada término se debe incluir E(C,termi) y GI(C,termi). Llamar al archivo ***mejores.txt***, con prefijo.

En resumen, faltan esas cosas

**Consideraciones finales**

La tarea puede ser realizada en grupos de dos personas.

La fecha de entrega es el lunes 3 de agosto al mediodía.

**STOPWORDS USADOS**

a, about, above, after, again, against, all, am, an, and, any, are, "aren't", as, at, be, because, been, before, being, below, between, both, but, by, "can't", cannot, could, "couldn't", did, "didn't", do, does, "doesn't", doing, "don't", down, during, each, few, for, from, further, had, "hadn't", has, "hasn't", have, "haven't", having, he, "he'd", "he'll", "he's", her, here, "here's", hers, herself, him, himself, his, how, "how's", i, "i'd", "i'll", "i'm", "i've", if, in, into, is, "isn't", it, "it's", its, itself, "let's", me, more, most, "mustn't", my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, "shan't", she, "she'd", "she'll", "she's", should, "shouldn't", so, some, such, than, that, "that's", the, their, theirs, them, themselves, then, there, "there's", these, they, "they'd", "they'll", "they're", "they've", this, those, through, to, too, under, until, up, very, was, "wasn't", we, "we'd", "we'll", "we're", "we've", were, "weren't", what, "what's", when, "when's", where, "where's", which, while, who, "who's", whom, why, "why's", with, "won't", would, "wouldn't", you, "you'd", "you'll", "you're", "you've", your, yours, yourself