Estructura de Datos I

Grado en Inteligencia Artificial

Objetivos

El objetivo de esta práctica es que el alumno se familiarice con el uso de árboles.

Enunciado

Elon Musk compró Twitter con el objetivo de hacerla más libre y fomentar la libertad de expresión que estaba en entredicho. Por ello, nada más llegar tuvo que hacer "*limpieza*" no solo de los empleados que trabajaban en la compañía, sino también en la estructura y funcionamiento de la plataforma.

Lo primero que analizó, él y su equipo, es que todos los hashtags publicados hasta la fecha se almacenaban en una lista. Elon no sabe de programación, tiene una intuición de que ese sistema es poco eficiente. Es por ello, que os ha contratado para que hagáis una pequeña prueba de concepto de un nuevo sistema de almacenamiento que clasifique los tweets (o post) en función de los hashtags que aparecen en ellos.

Como sabéis, un tweet es una publicación o actualización de estado realizada en la red social Twitter (o *X*). Un tweet es un mensaje de texto con extensión máxima de 140 caracteres (están permitidos letras, números, signos y enlaces). Los tweets pueden contener hashtags o etiquetas, que permiten establecer el tema del que trata el tweet. Hashtag es una palabra compuesta por hash (almohadilla) y tag (etiqueta) y son palabras que empiezan por el símbolo almohadilla (#), seguido de una palabra (por ejemplo: #programación, #cloud, etc.).

Para el desarrollo de esta práctica, y poder realizar la prueba de concepto, dentro del fichero `Material.zip` se pueden encontrar los siguientes archivos:

- `hashtags.txt`: archivo de texto donde cada línea tiene un hashtag.
- `tweets.txt`: que contiene un tweet en cada línea.

El etiquetado de tweets lo vamos a implementar como un árbol binario de búsqueda en el que cada nodo almacena un registro con dos campos:

- `hashtag`: que almacena un solo hashtag.
- `listaTweets`: que almacena una **lista dinámica** de tweets que contengan ese hashtag. La lista debe ser dinámica, ya que no conocemos a priori el número de tweets que van a contener cada hashtag. En este punto, puedes utilizar una implementación que ya tengas hecha.

Esta representación y la funcionalidad asociada deberá ir implementada en una unidad que se utilizará como el `tElemento` del árbol binario de búsqueda.

De modo que lo que se pide es:

1. *Construir la estructura fundamental del ABB*: para ello, leer la lista de hashtags proporcionada en hashtags.txt (cada tweet es una línea del fichero de texto) y construir el árbol binario de búsqueda de hashtags ordenado por orden alfabético. Para la construcción del ABB, recuerda que solo debes utilizar la interfaz de la unidad UArbolBB.
2. *Etiquetado de tweets*: Una vez construido el árbol, leer cada uno de los tweets almacenados en tweets.txt (cada tweet es una línea del fichero de texto). Para cada tweet, buscar los hashtags que contenga y almacenar el tweet en el nodo del ABB que corresponda al hashtag. Por ejemplo, un tweet como el siguiente "*Sería feliz estando todo el tiempo de #viajesPorElMundo*" debería estar almacenado en la listaTweets del nodo cuyo valor del campo hashtag coincida con "*#viajesPorElMundo*". Por otro lado, puede suceder que un tweet contenga más de un hastag.
3. *Consultas*: como resultado, hemos construido una base de datos de juguete y podemos implementar la funcionalidad que permita realizar consultas de tweets por hashtag. Es decir, el usuario podrá preguntar por un hashtag cualquiera y el programa deberá responder con la impresión por pantalla de los tweets que contienen el hashtag consultado o, en su caso, que no se ha encontrado ningún tweet con ese hashtag.
4. ¿Qué diferencias existen entre el sistema que habéis desarrollado y el modelo anterior que se encontró Elon Musk al llegar a Twitter?

NOTAS IMPORTANTES PARA REALIZAR LA PRÁCTICA

- Para poder identificar correctamente los hashtags, debéis tener en cuenta que empieza siempre por #, seguido de una secuencia de letras y números. Se considera que el hashtag ha terminado si tras la secuencia nos encontramos con un espacio o alguno de estos símbolos: `['\n', '.', ',', ' ', '?', '!', ':', ';' ]`
- Una vez identificado el inicio y el fin del hashtag, podremos copiarlo en un nuevo `strncpy`:

  ```
  char *strncpy(char *s1, const char *s2, size_t n);
  ```

  Por ejemplo, si en la cadena `tweet` hemos identificado un hashtag que se inicia en la posición 5, tiene una longitud de 10 caracteres y lo queremos copiar a un `char* destino`, la llamada sería:

  ```
  strncpy(destino, tweet+5,10);
  ```