TRABAJO ACADÉMICAMENTE DIRIGIDO

Diseño y contrucción de un prototipo para la extracción de información de redes sociales para su uso en sistemas de recomendación

Profesor: **D. José Carpio Cañada**

Alumno: **Luis Israel Diéguez Arjona**

ÍNDICE

1 INTRODUCCIÓN

El documento actual pretende ser una guía que sirva para conocer los detalles del TAD (en adelante, denominado como “proyecto”).

Éste ha sido dirigido por D. José Carpio Cañada, profesor de la Universidad de Huelva, y desarrollado por Luis Israel Diéguez Arjona, alumno de la misma universidad.

El proyecto ha contado con un total de 70 horas.

En la actualidad, D. José Carpio está trabajando en su tesis “Sistemas de recomendación”, en la que pretende aprovechar el etiquetado social en Internet. Ello puede servir de ayuda a la hora de buscar sitios web sobre una temática específica. Las recomendaciones se realizan de manera similar a la de Amazon: “Las personas que compraron este libro, también compraron estos otros...”

Para la recolección de datos en los que basar el sistema, se han tomado dos fuentes de información:

- Delicious
- Twitter

En versiones posteriores del sistema, se incluirá la recolección de datos de otras fuentes como Facebook. Todo apunta a que las redes sociales van a marcar el futuro de Internet, y en ellas va a jugar un papel fundamental el Data Mining. La interactividad del usuario va a primar en este futuro.

El desarrollo del proyecto que nos ocupa, pasa por el cumplimiento de los siguientes tres puntos:

- Captura de información
- Procesado de la misma
- Almacenamiento

La captura de datos ha tomado como base Twitter, haciendo uso de su API. Esta información ha sido procesada para extraer la información deseada, y darle un formato adecuado para, posteriormente, ser almacenada en una base de datos.

2 HERRAMIENTAS

Para el desarrollo del proyecto, hemos hecho uso de las siguientes herramientas:

- Twitter API
- PERL
- WGET
- MySQL

Comentar que, como sistema operativo, ha sido usado Linux en su distribución Ubuntu.

3 PREPARANDO EL EQUIPO

Para el desarrollo del proyecto, como bien hemos comentado anteriormente, hemos hecho uso de Ubuntu en su versión 10.10, con la instalación de algunos paquetes y módulos extra que detallamos a continuación.

Para ello nos hemos valido del gestor de paquetes Synaptic.

MySQL

Instalación de los paquetes siguientes:

mysql-server
mysql-client

Durante la instalación, se nos solicitará la introducción de los datos de acceso a la base de datos.

Apache2

Instalación del paquete:

apache2

Una vez finalizada la instalación, verificar su correcto funcionamiento yendo a la URL siguiente:

http://localhost

Aparecerá un mensaje como el siguiente:

It works!

PHP5

Para la instalación de PhpMyAdmin, el cual nos facilita mucho la tarea de administrar la base de datos, necesitaremos tener instalado PHP5.

Instalación de los paquetes siguientes:

php5
libapache2-mod-php5

Reiniciar el servidor apache2 desde la terminal:

sudo /etc/init.d/apache2 restart

Instalar los siguientes paquetes extra de PHP5:

php5-mysql
php5-curl
php5-gd
php5-idn

php-pear
php5-imagick
php5-imap
php5-mcrypt
php5-memcache
php5-mhash
php5-ming
php5-ps
php5-pspell

php5-recode
php5-snmp
php5-sqlite
php5-tidy
php5-xmlrpc
php5-xsl
php5-json

Volver a reiniciar el servidor apache2 desde la terminal:
sudo /etc/init.d/apache2 restart

Para verificar que todo quedo bien instalado, crear un archivo llamado
info.php

en el directorio
/var/www

con el siguiente contenido:

```
<?php
    phpinfo ();
?>
```

Abrir el navegador y acceder a la siguiente ruta:
http://localhost/info.php

Debería aparecer toda la información de la versión PHP instalada.

PhpMyAdmin

Instalación del paquete:
phpmyadmin

Durante el proceso de instalación, posiblemente se nos pida información adicional:
Web server to reconfigure automatically: apache2
Configure database for phpmyadmin with dbconfig-common? No

Para comprobar que PhpMyAdmin ha sido correctamente instalado, accedemos a:
http://localhost/phpmyadmin

Módulo PERL

XML :: Simple

Ejecutar en una terminal:

```
sudo perl -MCPAN -e 'install XML::Simple'
```

Net :: Twitter

Ejecutar en una terminal:

```
sudo perl -MCPAN -e 'install Net::Twitter'
```

4 FICHEROS

El proyecto consta de varios ficheros que pasamos a explicar a continuación:

dbCreate.sql
Contiene las sentencias SQL que deberemos ejecutar en la Base de datos para crear las tablas donde almacenaremos la información pertinente.

launcher.pl
Es el fichero encargado de llevar la cuenta de cuántas ejecuciones se realiza de la aplicación de captura, mandando a la misma vez a procesar los tweets. Nos mostrará un mensaje en cada iteración.

application.pl
Es el fichero que captura los tweets y los procesa. Es el encargado de interactuar con la API de Twitter, procesar los datos, y almacenarlos en la Base de datos.

tweets.xml
Fichero temporal que almacena los tweets obtenidos de la API y que han sido publicados desde la aplicación de Delicious.

tweetsUser.xml
Fichero temporal que almacena aquellos tweets obtenidos también desde la API y que pertenecen a un usuario específico. Sirve como apoyo a la búsqueda del usuario original de un tweet retweeteado.

5 CONEXIÓN OAUTH

Para tener la posibilidad de realizar un mayor número de peticiones al servidor de Twitter mediante su API, hemos usado el módulo Net :: Twitter de CPAN por su facilidad.

Para ello, hemos creado una nueva aplicación en el sitio web de la API de Twitter, en la siguiente dirección web:

http://dev.twitter.com/apps/new

Con ello podremos acceder a los datos fundamentales para poder realizar dicha conexión:

Consumer key
Consumer secret

Una vez realizada la conexión, podemos comunicarnos con los servidores de Twitter.

6 BÚSQUEDA Y CAPTURA DE INFORMACIÓN

Los búsqueda de información en Twitter ha sido posible gracias a su API.

Una API (Application Programming Interface) es, como su nombre indica, una interfaz de programación de aplicaciones. La API engloba una serie de funciones que serán usadas para el desarrollo de aplicaciones. Estas funciones pretenden ocultar lo máximo posible la complejidad de su funcionamiento interno.

Podríamos decir que la Twitter API pretende facilitar la comunicación entre Twitter y nuestra aplicación.

El sitio web de la Twitter API se encuentra en la siguiente dirección:
http://dev.twitter.com

Sin embargo, lo que más nos interesa de este sitio web es su documentación, localizada en la siguiente URL:
http://dev.twitter.com/doc

El uso que hemos hecho de la Twitter API ha sido el siguiente:

Obtención de los tweets publicados desde la aplicación de Delicious

Para ello vamos a hacer uso de los recursos de búsqueda
Search resources

que nos brinda la API.

Hemos comentado que los tweets que queremos obtener serán aquellos que hayan sido publicados desde la aplicación de Delicious. Para ello, haremos uso del parámetro
source:Delicious

Además, queremos especificar que contenga una dirección web. Como dicha dirección será una URL corta de Delicious, usaremos el parámetro:
q=icio.us

Para la descarga del fichero, nos hemos valido de WGET, una herramienta incluida en Ubuntu que permite la descarga de contenidos desde servidores web de una forma simple.

Obtención de los últimos tweets de un usuario específico

Para ello vamos a hacer uso de la herramienta

statuses/user_timeline

que nos brinda la API.

Deseamos poder acceder a los últimos tweets de un usuario en aquellos casos en los que detectemos que el tweet obtenido en la primera fase ha sido retweeteado (RT) por otro usuario. Nuestra intención es encontrar el tweet original, y para ello haremos un seguimiento que finalizará cuando se encuentre el tweet original o, en su defecto, el del primer usuario que lo retweeteó y al que podamos acceder.

Para ello, haremos uso de los parámetros siguientes:

screen_name: nombre del usuario del que queremos obtener los últimos 20 tweets.

count: número de tweets que queremos obtener. Si no hacemos uso de este parámetro, por defecto serán 20 tweets. En nuestro caso vamos a solicitar los 100 últimos tweets del usuario.

7 TRATAMIENTO DE LA INFORMACIÓN

Consiste en el procesamiento del fichero tweets.xml. Para ello usaremos el módulo de PERL “XML::Simple”.

Debido a que el procesado es una tarea de difícil explicación, únicamente se mencionará a un nivel superficial, dejando a un lado los detalles, que pertenecen a un nivel de programación.

Por cada tweet (referido al conjunto de toda la información perteneciente a un tweet, no al texto en sí), obtenemos el texto del tweet, la fecha de publicación, y la URI del usuario.

Para cada conjunto de esta información:

- Comprobar si es un RT.
  En tal caso, buscamos los últimos tweets del último usuario que aparezca en el tweet inicial con el formato “RT @usuario”.
  Sus tweets son comparados mediante el algoritmo de la distancia de Levenshtein (Levenshtein Distance Algorithm).
  Si resulta que dicha función devuelve TRUE, consideramos que hemos encontrado el tweet original. Será este, y no el leído inicialmente, el que almacenaremos en la base de datos, junto con su información asociada.
- Capturar los enlaces existentes.
- Capturar las etiquetas usadas en el tweet.
- Capturar las menciones de usuarios realizadas.

8 ALMACENAMIENTO DE LA INFORMACIÓN

La Base de datos está compuesta por una serie de tablas que vamos a ver a continuación:

tw_users
Contiene aquellos usuarios de los que hemos obtenido tweets. Almacena el nombre de usuario, su nombre real, y la URI.

tw_tweets
Contiene los tweets obtenidos. Almacena el nombre del usuario que ha publicado el tweet, el texto del tweet, y la fecha de publicación.

tw_links
Contiene los enlaces que había en el tweet. Almacena el link, el md5 de dicho link (para facilitar su comparación) y un contador que se incrementa si el link ya existía.

tw_links_tweet
Enlaza links con tweets. Cada link aparece en, al menos, un tweet. Esta tabla hace dicho enlace. Almacena el nombre del usuario que publicó el tweet, el texto del tweet, y el md5 del link.

tw_tags
Contiene aquellas etiquetas publicadas en los tweets. Almacena la etiqueta junto con un contador.

tw_tags_tweet
Enlaza etiquetas con tweets. Almacena el nombre del usuario, el texto del tweet, y la etiqueta.

tw_ats
Contiene aquellas menciones que se hacen en los tweets. Almacena el nombre del usuario que es mencionado, junto con un contador.

tw_ats_tweet
Enlaza menciones con tweets. Almacena el nombre de usuario, el texto del tweet, y el nombre del usuario al que se menciona en el tweet.

tw_statistics
Contiene datos estadísticos. Almacena, para cada bloque de inserciones, los siguientes datos:

Tweets: total de tweets capturados mediante la API, total de tweets insertados, y total de tweets ya existentes en la base de datos.

Users: total de usuarios que publicaron los tweets capturados, total de usuarios insertados, y total de usuarios ya existentes en la base de datos.

Links: total de enlaces encontrados entre los tweets obtenidos, total de enlaces insertados, y total de enlaces ya existentes entre los registros de la base de datos.

Tags: total de etiquetas encontradas entre los tweets, total de etiquetas insertadas, y total de etiquetas ya existentes en la base de datos.

Ats: número de menciones que se realizan en los tweets, totalidad de usuarios mencionados insertados, y totalidad de usuarios mencionados ya existentes en los registros de la base de datos.

Averages: media de usuarios por tweet, de enlaces por tweet, de etiquetas por tweet, y de menciones por tweet.

Others: hora de almacenamiento, tamaño del fichero (actualmente en desuso), nombre del fichero (también en desuso en la actualidad).

9 PUESTA EN MARCHA DE LA APLICACIÓN

Para hacer trabajar la aplicación, no tenemos más que seguir los siguientes pasos:

1. Iniciar la base de datos (en caso de no tenerla configurada para que se inicie automáticamente)

2. Ejecutar el lanzador (launcher.pl)

Con estos dos sencillos pasos, la aplicación trabajará sola:

- Captura de tweets
- Procesado
- Almacenamiento.

10 CONCLUSIÓN

En primer lugar, agradecer a D. José Carpio el haberme presentado en su momento la posibilidad de realizar este Trabajo Académicamente Dirigido, y el haber confiado en mí. Agradecerle también la cercanía que muestra con el alumno, y la flexibilidad con la que me ha permitido llevar a cabo el proyecto.

Ha sido un cuatrimestre duro por diferentes circunstancias, entre las que priman la gran cantidad de trabajo que he tenido que llevar adelante. Sin embargo, he de reconocer que éste proyecto ha servido para poner en práctica capacidades que, durante el transcurso de la carrera, no se nos exige.

Con el desarrollo de este proyecto he podido aprender un manejo más profundo de Linux, algo que deseaba desde hace algún tiempo. También he podido aprender un nuevo lenguaje de programación, PERL, y el uso de expresiones regulares, lo que dota a éste lenguaje de un gran potencial.

He de decir que ha sido un proyecto en el que he podido aprender nuevos conocimientos, a la vez que he disfrutado desarrollándolo.