

LECTOR Y RECOMENDADOR DE NOTICIAS

Acceso Inteligente a la Información

Curso 2018/2019

DOCUMENTACIÓN

Antolín Bermúdez, Miguel Ángel

De la Torre Díaz, David

Rodríguez Aránega, Cristian

1. DESCRIPCIÓN DEL PROBLEMA	3
2. DESCRIPCIÓN DE LA SOLUCIÓN	3
3. SCRAPING CON BEAUTIFUL	3
4. MODELADO DE DATOS CON DJANGO	4
6. EJECUCIÓN Y PUESTA EN MARCHA DEL SISTEMA	5
7. CONCLUSIONES	5

1. DESCRIPCIÓN DEL PROBLEMA

Hoy en día existen muchas páginas de noticias online, entre ellas se encuentran el nicho de los diarios web. Páginas donde se recogen las noticias actuales y los usuarios pueden dejar comentarios y opiniones.

No obstante no existe una página la cual unifique todos los diarios para recoger la información actual y no tener que acceder página a página buscando noticias que nos puedan resultar interesantes.

En el siguiente punto se describe la solución realizada por el grupo con más detalle.

2. DESCRIPCIÓN DE LA SOLUCIÓN

Para crear la solución a este problema se ha creado un programa *demo* en el cual se realiza *scraping* a tres diarios web, en ellos recogemos el titular de la noticia, su autor, su fecha y su enlace. Para un programa en su versión final el *scraping* realizaría de muchos más diarios *online*, no obstante dado el poco tiempo disponible se ha realizado únicamente el *scraping* de tres diarios.

Una vez obtenida esta información, presentamos una página web con un menú el cual permite al usuario visualizar todas las noticias del día (Noticias de las portadas de los tres diarios), buscar noticias por palabras clave y un apartado capaz que dada una noticia que el sistema te recomiende cinco noticias parecidas / relacionadas a la inicial.

El sistema de recomendación utilizado se trata de un sistema basado en contenido, en el cual a través de los títulos de las noticias busca similitudes entre ellas y crea un top cinco noticias más parecidas a la seleccionada. Al no tener puntuaciones de usuarios este era la única metodología factible para desarrollar esta solución.

3. SCRAPING CON BEAUTIFUL

Para la obtención de los datos sobre las noticias necesarios para la implementación del sistema hacemos Scraping con BeautifulSoup.

Para ello se han creado tres archivos python, uno para cada web a rascar, en el cual se incluyen los enlaces de acceso a el repositorio de noticias, y se obtienen los datos según el formato de cada una de estas.

Así obtenemos una tupla por cada noticia, de la forma:

```
tupla = (tituloNoticia,enlaceNoticia,autorNoticia,comentariosNoticia,fechaNoticia)
```

Que serán las tuplas que se añadan a la base de datos para crear la tabla en *mysql/3* de *entrega_noticias*.

También se crea una tabla Autor con los correspondientes autores de las noticias.

Toda esta información se recopila al ejecutar el archivo *populate_database.py* en el siguiente punto.

4. MODELADO DE DATOS CON DJANGO

Para poder almacenar todos los datos obtenidos en el paso anterior y crear una base de datos *mysql/3* compatible con *Django*, hemos creado el archivo *populate_database.py*, que se ejecutará al hacer click en el enlace de “cargar” del menú principal de la aplicación.

Para ello se han creado los modelos Autor, con un *idAutor* y un *nombre*; y Noticias, con *IdNoticia*, *titulo*, *link*, *autor*, *comentarios* y *fecha*.

Para las tuplas con el formato creadas anteriormente,

```
tupla = (tituloNoticia,enlaceNoticia,autorNoticia,comentariosNoticia,fechaNoticia) se
```

procede a recorrer cada una de ellas y se insertan en la base de datos, creando primero los Autores y una vez creados, creando referencias con cada una de las noticias que han publicado.

5. SISTEMA DE RECOMENDACIÓN BASADO EN CONTENIDO

Dado el título de cada noticia, se ha creado un sistema que por medio del cálculo de la similaridad del coseno, proporciona una puntuación para cada noticia dada su similaridad con la noticia a comparar.

Para ello creamos un archivo csv temporal que incluye todas las noticias de nuestra base de datos mysql3, y lo transformamos en una matriz TF-IDF con los títulos de cada noticia, de las cuales eliminamos las palabras STOP, para dotar a nuestro algoritmo de mayor eficiencia y rapidez.

De esta matriz, calcularemos su núcleo polinomial y obtendremos así las similitudes por coseno y las puntuaciones para recomendar.

Al hacer una recomendación, el sistema devolverá por terminal cada una de las películas recomendadas junto a su puntuación recomendada, por ejemplo:

- Los Mossos investigan la muerte de un detenido en una comisaría de Barcelona (puntuación:0.08413261778957559)
- Sin justicia para las víctimas del 'doble crimen del Cash Récord' (puntuación:0.04705005105424572)
- Interior compró una pistola al chófer de Bárcenas (puntuación:0.031598957785416)

Una noticia recomendada con una puntuación mayor a 0.1 resulta ser una predicción de recomendación bastante acertada.

6. EJECUCIÓN Y PUESTA EN MARCHA DEL SISTEMA

1. Instalar requirements.txt con el comando:

- pip3 install -r requirements.txt

2. Realizar las migraciones en este orden:

- python3 manage.py makemigrations entrega
- python3 manage.py migrate

3. Ejecutar el programa

- python3 manage.py runserver

4. Acceder a la dirección web: <http://localhost:8000/>

5. Cargar datos en el sistema haciendo click en el enlace “Cargar toda la información” del menú.

6. Ejecutar cualquiera de los métodos del menú presentados en *localhost*.

7. CONCLUSIONES

Hoy en día hay muchas páginas que muestran información entre ellas páginas de diarios web. Se encuentra mucha información repartida en múltiples páginas, nosotros hemos juntado toda esta información en un único lugar.

Además hemos conseguido recomendar noticias similares entre ellas utilizando un sistema de recomendación.

Hemos aprendido a realizar un sistema basado de recomendación basado en contenido únicamente utilizando el título de las noticias. En el sistema actual se encuentran 88 notcias entre sí y tarda pocos segundos en encontrar las noticias similares.

