Proyecto 2021-2022

Búsqueda y recomendación de texto

Objetivo

Implementar en grupos de trabajo un programa original en Python que realice las siguientes tareas:

1. Descargar un conjunto de noticias de diferentes medios en Internet mediante web scraping.
2. Permitir realizar búsquedas sobre esta base de documentos utilizando técnicas de recuperación de información sobre texto.
3. Ofrecer recomendaciones basdas en contenido a partir de una noticia preselecionada.

Descripción

La aplicación deberá emplear técnicas de **web scraping** para descargar el **texto** de noticias de categorías prefijadas. Las fuentes serán diferentes medios de noticias en Internet.

Para cada sección, deberán descargarse todas sus noticias para almacenarse en formato texto.

A partir de la base de documentos generada, el programa deberá permitir tres acciones a través de su interfaz gráfica:

1. Realizar una búsqueda de texto libre.
2. Seleccionar una noticia y buscar y mostrar las más similares a ella.
3. Seleccioanr una noticia y recomendar otras en base a su contenido.

En todos los casos el resultado será un ranking de los N documentos que mejor se ajusten a la búsqueda/recomendación en cuestión (inicialmente N=5). Se mostrarán los nombres de los ficheros de texto, junto con su grado de similitud. Deberá poderse seleccionar el recurso para poder leer el texto.

En los casos 2 y 3, la noticia se seleccionará a partir de un directorio (medio → categoría → noticia). Deberá igualmente poderse leer.

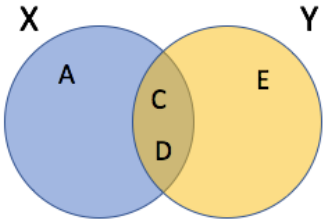
Para realizar la recuperación de texto (búsquedas) se utilizará el **modelo vectorial**, pre-procesando el texto junto con una lista de parada. Se calculará la **similitud** mediente el ángulo del **coseno** de los vectores, y para determinar los pesos de los términos en los documentos y la consulta (query) se empleará **TF-IDF**.

Para cubrir el punto 3, utilizaremos las **keywords** (etiquetas / tags) establecidas en las noticias para definir su contenido, y calcularemos la similitud en función del coeficiente de Sørensen–Dice.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

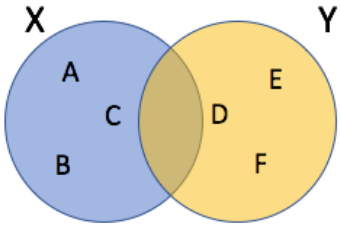
A grosso modo, este enfoque permite evaluar cuán parecidos son dos conjuntos en función del **número de elementos que tienen en común**. El resultado será un número entre 0 y 1 (similitud nula o similitud total, respectivamente).

A continuación figuran algunos ejemplos a modo de ilustración:



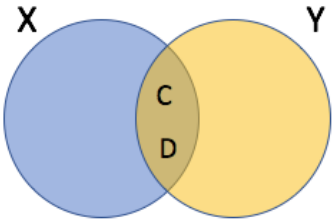
$$Sim\,(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|} = \frac{2 \times |\{A,C,D\} \cap \{C,D,E\}|}{|\{A,C,D\}| + |\{C,D,E\}|}$$

$$= \frac{2 \times |\{C,D\}|}{3+3} = \frac{2 \times 2}{6} = \frac{4}{6} = 0{,}66$$



$$Sim\,(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|} = \frac{2 \times |\{\emptyset\}|}{|\{A,B,C\}| + |\{D,E,F\}|}$$

$$= \frac{2 \times 0}{6} = \frac{0}{6} = 0$$



$$Sim\,(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|} = \frac{2 \times |\{C,D\} \cap \{C,D\}|}{|\{C,D\}| + |\{C,D\}|}$$

$$= \frac{2 \times |\{C,D\}|}{2+2} = \frac{2 \times 2}{4} = \frac{4}{4} = 1$$

Cada uno de los medios consultados dispone de su propio **sistema de etiquetado de contenido**. Este será el que emplearemos para caracterizar cada noticia. En este sentido, **nuestro sistema será dependiente** por completo de lo buenas o malas que sean las etiquetas. Tenemos situaciones en las que apenas se aportan etiquetas, y otras en las que son bastante minuciosas (figura 1).

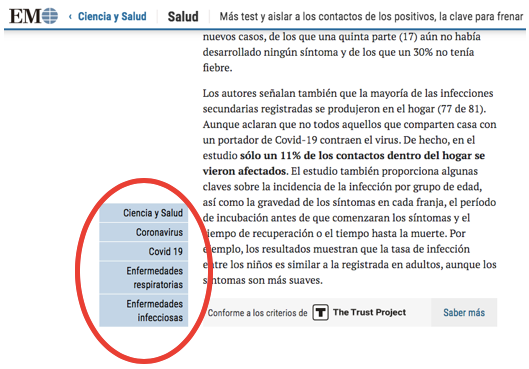


Figura 1. Etiquetas en las noticias

Medios y categorías

Se trabajará con los siguientes diarios online y categorías:

- El Mundo (https://elmundo.es)
  - Salud: https://www.elmundo.es/ciencia-y-salud/salud.html
  - Tecnología: https://www.elmundo.es/tecnologia.html
  - Ciencia: https://www.elmundo.es/ciencia-y-salud/ciencia.html

- El País (https://elpais.com/)
  - Sanidad: https://elpais.com/noticias/sanidad/
  - Tecnología: https://elpais.com/tecnologia/
  - Ciencia: https://elpais.com/ciencia/
- 20 Minutos (https://www.20minutos.es/)
  - Salud: https://www.20minutos.es/salud/
  - Tecnología: https://www.20minutos.es/tecnologia/
  - Ciencia: https://www.20minutos.es/ciencia/

Almacenamiento

Las noticias deberán almacenarse en ficheros de texto. Cada noticia tendrá el nombre de su categoría, seguido de la fecha (aaa-mm-dd) y un número que indicará su orden dentro del listado de ese día (nnn). Para evitar confusiones, cada texto estará almacenado dentro de una carpeta con el bombre de la categoría, y esta a su vez en una carpeta con el nombre del medio desde el que se descargó. Un ejemplo para el caso de la categoría de “salud” del 20minutos, podría ser el siguiente :

20Minutos

Salud

salud.2020-03-14.001.txt

salud.2020-03-14.002.txt

…

salud.2020-03-16.004.txt

Sugerencia de interfaz

Búsqueda de noticias

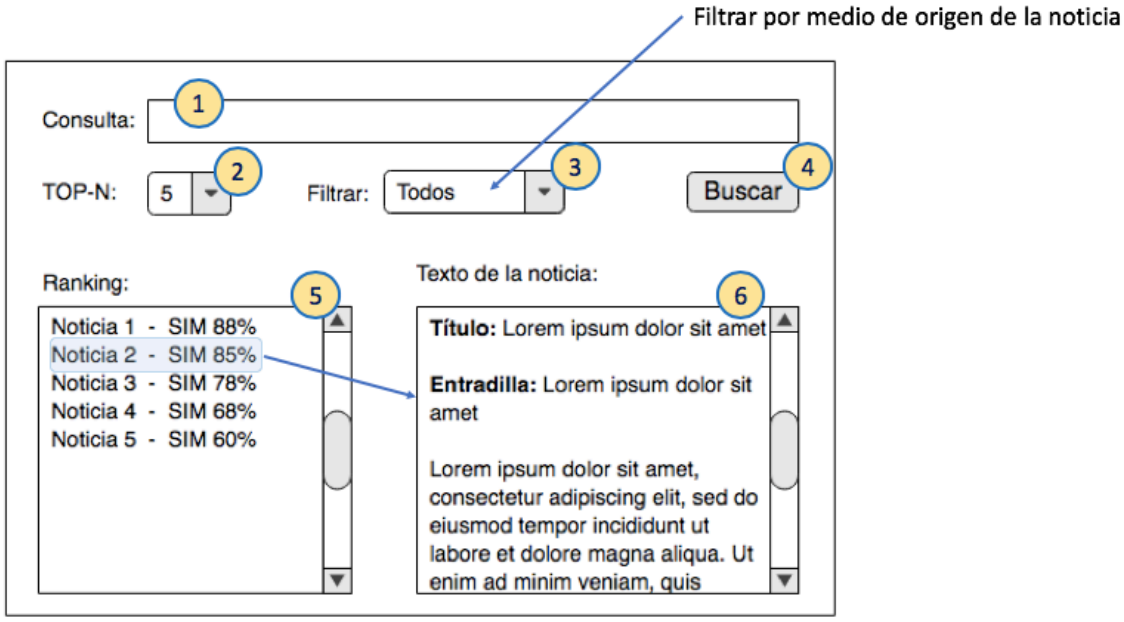


Figura 2. Sugerencia de inerfaz para buscar en una noticia

1. Introducir consulta.
2. Seleccionar número de ítems (noticias) en el ranking.
3. Filtrar las noticias por el medio de origen o seleccionar todos.
4. Buscar
5. Mostrar las noticias por roden de similitud.
6. Ver el texto de la noticia seleccionada.

Sugerencia de interfaz

Comparación de noticias / Recomendación de noticias

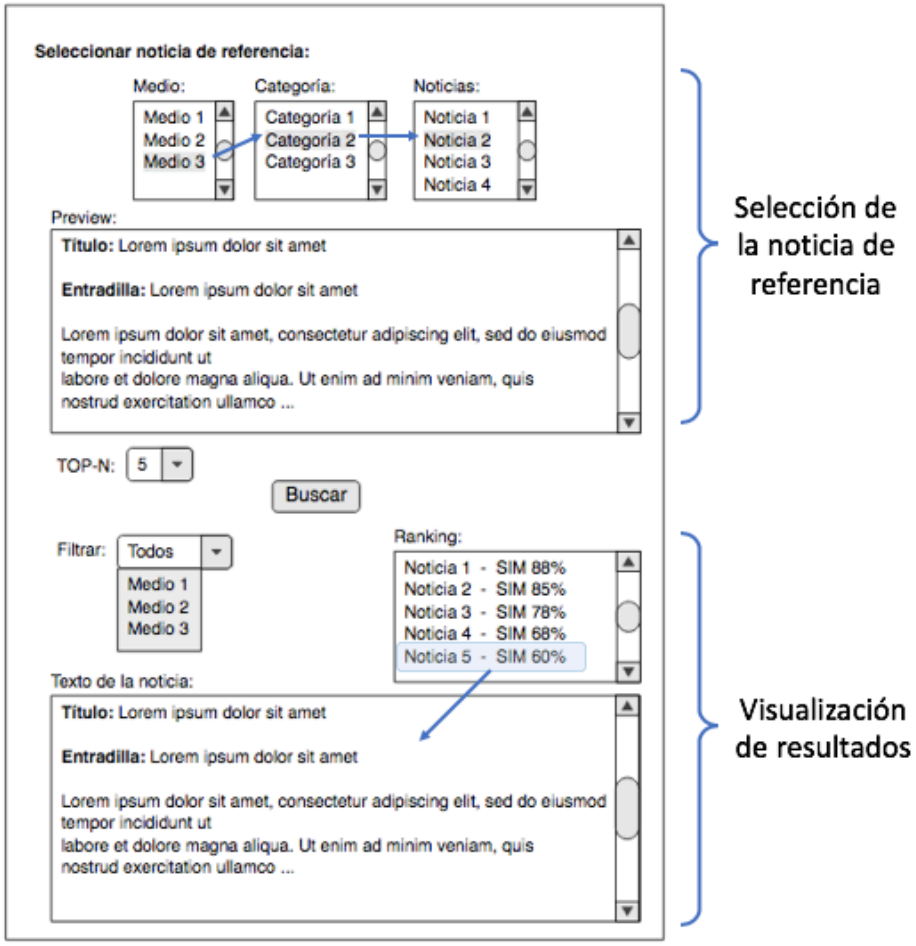


Figura 3. Sugerencia de inerfaz para buscar/recomendar en base a una noticia

Documentación

La documentación de esta primera fase se adjuntará en un fichero PDF, en el que dse describirá el trabajo realizado.

El esquema a seguir podría ser similar al siguiente:

- Portada (titulación, asignatura, actividad, estudiante, fecha, etc.)
- Índice
- Introducción (descripción del problema a resolver y enfoque se se va a seguir, qué es el webscraping, el modelo espacio vectorial, TF-IDF, el áclculo de la similitud en base al coseno, etc.)
- Descripción / organización del proceso:
    - Organización del código: clases, principales fuciones/métodos, librerías paquetes utilizados (¿por qué?), etc.
    - El proceso de webScraping.
        - Fuentes de datos (El Mundo, 20Minutos, El País) y categorías.

        - Análisis de las fuentes de datos: cómo se han extraído los datos desde el HTML.
    - El proceso de recuperación de texto:
        - Almacenamiento de los textos
        - Cálculo de las similitudes (en base a query y en base a otras noticias)
    - El proceso de modelado: creación de vectores y ponderación.
    - El prpeso de recomendación de texto.
- Pruebas
- Resultados
- Conclusiones
- Apéndices:
    - Manual de instalación.
        - Especal énfasis en la configuración del entorno Python para poder replicar el proyecto.
    - Manual de usuario.
- Bibliografía.

Normas de entrega

La entrega consistirá en un único archivo comprimido en **formato ZIP** (no se aceptan entregas en RAR, 7Z u otros formatos).

Seguir las siguientes indicaciones para adjuntar el envío a través del Campus Virtual. No cumplir estas normas supondrá la no aceptación de la entrega:

1. Crear una carpeta cuyo nombre seguirá el siguiente formato: **Grupo_X.SSII.Proyecto1**, donde la **X** representa el número del grupo.
2. Dentro de la anterior carpeta, incluir:
    a. Una subcarpeta llamada “CODIGO”, que incluirá:
        i. Los ficheros fuente en Python necesarios para la ejecución del programa.
        ii. La estructura de carpetas y ficheros TXT necesarios para el correcto funcionamiento del proyecto. Deberá haber **entre 10 y 20 ficheros de texto** por cada categoría de cada medio.
    b. El documento en **formato PDF** con todos los detalles de la actividad. Su nombre seguirá el siguiente formato:

        Grupo_X.SSII.Proyecto1.**PDF**

3. Comprimir en **formato ZIP** la carpeta raíz contenedora de todos los ficheros, especificada en el punto 1. El nombre deberá seguir el siguiente formato:

    Grupo_X.SSII.Proyecto1.**ZIP**

Deberá subirse al Campus Virtual a través de la actividad correspondiente **antes de las 23:55h** de la fecha marcada como límite.