SIW PL5. Creación de un índice.

Hugo Fonseca Díaz (UO258318)

Table of Contents

Requerimientos

Se necesita Python 3.8 y la biblioteca *nltk* para ejecutar el script *example.py* que a su vez utiliza otro script denominado *index.py*. También se requiere el tener la colección *cran-1400.txt* en la misma carpeta.

Detalles de la implementación

El script *index.py* está compuesto por dos clases:

- **Index:** clase principal. Modela el fichero invertido y es la que interactuaría con otros scripts que quisieran usar el índice. Tiene la siguiente interfaz:
  - ***load_document(self, id_doc, document):*** carga un documento convirtiéndolo a una bolsa de términos que se insertan al índice.
  - ***put(self, term, entry):*** inserta un término con su correspondiente entrada al índice.
  - ***update_post_list(self, term, id_doc, tf):*** inserta un nuevo id de un documento junto a la frecuencia de término correspondiente al mismo a la *post-list* de un término que ya existe en el índice.
  - ***get_post_list(self, term):*** devuelve la *post-list* de un determinado término.
  - ***get_idf(self, term):*** devuelve la *inverse document frequency* de un determinado término.

    - ***get(self, term):*** devuelve la entrada asociada a un término del índice.

- **IndexEntry:** modela una entrada del índice. Contiene la *post-list* del término y una referencia al mismo. Tiene la siguiente interfaz:
    - ***get_idf(self, n_documents):*** calcula la *inverse document frequency* del término recibiendo el número de documentos actuales en el índice.
    - ***get_post_list(self):*** devuelve la *post-list* asociada al término.
    - ***update_post_list(self, id_doc, tf):*** añade una nueva entrada a la *post-list* asociada al término.

Aclaraciones y notas adicionales

El propio índice convierte documentos a bolsas de palabras dando un id del documento y el texto que lo comprende. Los términos de éstas bolsas de palabras son luego introducidos al índice por medio de la función *put*. Solo se puede introducir un documento a la vez. Si un término ya está incluido en el fichero invertido simplemente se actualiza la *post-list* de su entrada asociada mediante la función *update_post_list*.

Asumiendo que se le pueden introducir documentos en cualquier momento al índice, se pasa el número de documentos actuales al método que calcula la *idf* del término en la clase que modela una entrada del índice.

Entre los ficheros entregados se encuentra uno llamado *result_example_supersonic.txt* con el output de la ejecución del comando:

$ python example.py -t supersonic

Si se quisiera ver la ayuda de este script puede consultarse mediante el comando:

$ python example.py -h