PRACTICA 1: Extracción y búsqueda de información textual (3 puntos)

Para esta práctica se requiere al estudiante emplear técnicas de búsqueda y recuperación de la Información sobre datos no estructurados. En concreto, el estudiante aprenderá a compilar datos en la web, almacenarlos debidamente, extraer información y programar un buscador que trabaje tanto con el texto del cuerpo de los documentos como sus metadatos.

Generalmente, los datos no-estructurados se pueden obtener mediante web crawlers y desde redes sociales que faciliten APIs para el acceso a los contenidos (como podría ser en el caso de Reddit, Youtube o Twitter. Para esta práctica, nos centraremos en el uso de uno o varios web crawlers sencillos para extraer contenido desde uno o varios portales web, en base a un dominio elegido por el usuario, como pueden ser economía, deportes, cocina o música.

Para realizar esta práctica se debe usar un cuaderno de Jupyter (Jupyter Notebook) y puede usarse Google Colab como plataforma para ejecutarlo. En este cuaderno debe explicarse la resolución de cada uno de los apartados y aportar el código necesario para su correcta ejecución.

En concreto, esta práctica 1 se divide en los siguientes apartados:

Parte 1- Compilar datos de documentos web (0,75 puntos)

El primer objetivo en esta práctica consiste en extraer un número significativo de documentos web en español. Los estudiantes pueden escoger la temática como se ha comentado anteriormente. El conjunto de documentos que se compilen debe de ser almacenado en disco en ficheros de texto independientes. De cada documento deben de almacenarse los campos de título, contenido, fecha. Se recomienda el formato JSON.

Este sería un ejemplo del fichero de texto en JSON con la información extraída:

```
{
    "date": "2021-09-13",
    "document_type": "web",
    "url":      "https://www.sportcartagena.es/articulo/fc-cartagena/real-madrid-predica-ejemplo-cobra-abonos-ayuda-socios-mas-veteranos-conseguir-entrada-internet/20210908135004099428.html",
    "title": "Los blanquinegros bajan de golpe cinco escalones y se quedan en el puesto 14",
    "content": "La cal y arena que está dando el Cartagena en estos primeros compases ligueros lo están llevando por un auténtico carrusel de emociones combinando buenos partidos –en realidad solo el de la victoria ante la Real B- con partidos malos y otros, como el de Oviedo, malísimos.\nEl Cartagena ya ha salido del TOP-10 donde solo ha podido estar una jornada (por ahora). Habrá que confiar en que empiecen a remontar ya y que el equipo demuestre la calidad que se dice que tiene. Con solamente tres goles en cinco jornadas no da para mucho más."
}
```

Es importante hacer notar que estos ficheros de texto se podrán ir modificando más adelante conforme se extraiga más información del texto del documento mediante otras herramientas.

Para ello se utilizará la herramienta Scrapy https://scrapy.org/, Los documentos extraídos

deberán almacenarse para su posterior entrega en esta práctica. Recomendamos que se extraiga un número importante de documentos (al menos 500 documentos) realizando el crawling de distintos sitios web como noticias, blogs, etc. Estos nuevos Crawlers Web deben ser distintos de los vistos en clase y al menos uno de ellos debería ser un crawler de contenidos en HTML y no solamente utilizar el crawler de JSON-LD visto en clase.

También se podrán obtener documentos de otras fuentes como RSS o sitios que publiquen la información en JSON-LD, pero es necesario realizar al menos un crawler HTML.

Además del código fuente para este apartado se tendrán que entregar los archivos .json en un archivo ZIP.

Parte 2 – Buscador (0,75 puntos)

El siguiente paso es el de programar un buscador para localizar la información almacenada en los ficheros. Para ello, haremos uso de Elasticsearch como vimos en clase y de su API en Python para crear registros y realizar búsquedas (https://www.elastic.co/guide/en/elasticsearch/reference/current/search-your-data.html). La idea es programar un buscador que incluya al menos las siguientes características:

1. Búsqueda con operadores lógicos (AND, OR, NOT)
2. Búsqueda de resultados entre rango de fechas
3. Establecer el orden de los resultados
4. Resaltar las frases que coincidan con los criterios de búsqueda
5. Paginación de resultados
6. Uso de stemming
7. Búsquedas fuzzy (Fuzzy queries)
8. Uso de term suggester por si hay errores ortográficos en las búsquedas

Se pondrán ejemplos de distintas búsquedas con las características arriba mencionadas en celdas de código de Jupyter Notebook.

Además, se entregará en esta parte también el índice exportado en un fichero .json como se vio en la sesión de prácticas 3 de Elasticsearch.

Parte 3 – Vectorización (0,5 puntos)

En esta tarea, vamos a convertir en vectores los términos en el documento y vamos a calcular los términos con mayor ponderación usando TF/IDF.

Para ello, se debe usar la librería scikit-learn (http://scikit-learn.org/stable/). En concreto, familiarizaros con scikit-learn (http://scikit-learn.org/stable/) y, en particular, sus posibilidades para extraer características a partir de texto (sección 6.2.3 de la página https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction) y el TfIdf Vectorizer https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer

En esta parte será necesario hacer lo siguiente:

1. Obtener una matriz de documentos y términos.
2. Obtener cuáles son los 10 términos más relevantes de toda la colección (los que tienen mayor TF-IDF a lo largo de todos los documentos)
3. Se filtrarán términos mediante stop-words y eliminando aquellas palabras que

aparezcan en menos de 10 documentos).
4. Adicionalmente, hay que mostrar los 100 términos más repetidos en la colección.

Parte 4 Detección de entidades (opcional 0,5 puntos)

Se empleará la herramienta Stanza (vista en clases de prácticas) para obtener un listado de entidades según su tipo. Estas entidades se añadirán el fichero JSON en distintos campos según su tipo y se modificará la indexación del buscador de la parte 2 para poder buscar también por estos campos.

Debido a que este proceso puede tardar mucho tiempo, se podrá realizar sobre un subconjunto de 100 documentos de los extraídos en la primera parte.

Se deberán entregar los ficheros .json modificados, así como el nuevo índice de Elasticsearch exportado.

También se deberá incluir alguna celda en el Jupyter Notebook para buscar por las entidades.

Parte 5 (opcional 0,5 puntos) - Extracción de tópicos

Utilizad Gensim para realizar una extracción no supervisada de tópicos existentes en el corpus extraído, así como para su posterior visualización.

Referencias relacionadas:
GENSIM. https://radimrehurek.com/gensim/

Topic Modeling with Gensim. Example with LDA:
https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

Entregables:

1) Scripts de Python (.py)
2) Python Notebooks (.pynb)
3) Enlace para la descarga de los documentos generados (.zip)

Es fundamental que el Notebook sea auto-explicativo en todos los pasos (con celdas textuales acompañando a celdas con código y que contenga explícitamente los resultados -sin tener que ejecutar las celdas de nuevo-). Es recomendable en el Notebook incluir algunos ejemplos del funcionamiento de cada apartado. Por favor, comprobad esto antes de enviar el Notebook. Cualquier proyecto de Analítica de Datos debe ser autodocumentado y sus experimentos fáciles de reproducir. Un aspecto clave en la evaluación de esta práctica reside en la calidad de las explicaciones y documentación que acompañéis al código dentro del Notebook.

Valoración y Fecha de Entrega:

- Esta práctica tiene una valoración de 3 puntos (sobre el total de 7 puntos de la parte práctica de la materia). Dos puntos se corresponden a la correcta realización de los apartados obligatorios -apartados de 1) a 3)- y el punto adicional se corresponde con la correcta realización de los apartados optativos.
- La práctica se entregará a través de la tarea asignada en el AulaVirtual

Fecha límite entrega: 20 de enero a las 23:55h