Tecnologías de Gestión de Información No Estructurada
Master Universitario en Tecnologías de Análisis de Datos Masivos: Big Data

PRÁCTICA 1: Extrayendo y analizando textos procedentes de una Red Social

La web (ya sea una red social o una página web) representa una importante fuente de acceso a contenidos de "Big Text Data". En este primer trabajo práctico, nos centraremos en extraer contenido de una red social de referencia a nivel mundial: Reddit. Este trabajo está orientado a familiarizarse con técnicas básicas de creación de corpus textuales, así como en una primera exploración del contenido de los textos mediante la extracción de términos centrales o importantes.

Pasos a seguir:

1) El primer objetivo de la práctica consiste en extraer **datos** de Reddit acerca de un tema que os resulte de interés (un artista, una serie, literatura, un deporte, etc). Para ello, deberéis escoger al menos una subcomunidad de Reddit ("subreddit") donde haya suficientes interacciones entre los usuarios de modo que exista en la misma un buen número de publicaciones y comentarios. Esta parte es libre y abierta y podéis extraer documentos de cualquier comunidad Reddit que os resulte de interés. Una comunidad que genera debate y comentarios entre los usuarios es más interesante que una comunidad que es una mera secuencia de enlaces a sitios externos. La extracción debe estar centrada en comunidades cuyas interacciones entre usuarios sean realizadas en Inglés.

   El resultado de este paso debe ser un corpus normalizado consistente en una colección de publicaciones. El objetivo es crear un **dataset de,** por lo menos, **varios miles de entradas**.

   Para la extracción de datos podéis utilizar librerías de Python como:
   - Praw: https://praw.readthedocs.io/en/stable/
   - Alien: https://github.com/labteral/alien

   En el notebook a entregar debe aparecer todo el código asociado a la extracción de contenidos y parseado para la creación del corpus. No sería válido obtener una colección de terceros y usarla para las partes posteriores del proyecto. Es necesario que cada alumno/a trabaje en la extracción de los datos a partir de al menos una subcomunidad Reddit.

2) El corpus que obtengáis en el paso anterior debe ser almacenado en disco en un formato adecuado. Para ello, definid un esquema **JSON** o **XML** que permita almacenar toda la información disponible (guardando al menos título y contenido de cada publicación; se recomienda incorporar campos para todos los datos disponibles, por ejemplo no sólo título y contenido del texto sino también guardando el/la usuario/a que hace el escrito, subcomunidad o foro donde se publicó, fecha, etc.). Guardad toda la colección en un único fichero. Estos archivos deben ser legibles desde código Python utilizando, por ejemplo, la API ElementTree XML (para XML) o una biblioteca análoga para el procesamiento de JSON.

3) Realizad un sencillo tratamiento inicial del corpus anterior para vectorizar la colección, produciendo una representación "sparse", y mostrando los términos más ponderados por **tf/idf**. Para ello:

(a) instalad y familiarizaos con scikit-learn (http://scikit-learn.org/stable/) y, en particular, con sus posibilidades de extraer características del texto (sección 6.2.3 en la página https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction) y con el vectorizador TfIdf: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer

(b) dado el corpus obtenido de Reddit, y considerando cada entrada o post como un documento individual, utilizad el vectorizador TfIdf (filtrando las *stopwords* y todas aquellas palabras que aparezcan en menos de 10 docs) para vectorizar la colección y luego mostrar los 50 términos más "centrales" de la colección. Entendiendo como más centrales aquellos cuya suma acumulada de tf/idf sobre todos los documentos es mayor. Además, mostrar también los 100 términos más repetidos de la colección (suma de su tf o *term frequency* en los documentos).

4) En este paso, se os pide volver a sacar los términos más relevantes de la colección. Pero, esta vez, procesando cada texto con un modelo neuronal que te produce palabras clave asociadas a cada texto. En concreto, utilizaremos una técnica neuronal con *embeddings* de un modelo avanzado denominado BERT. Esta técnica representa los documentos y términos en un espacio vectorial denso. Debéis utilizar la librería Python **KeyBERT:** https://github.com/MaartenGr/KeyBERT. Cada texto del corpus os producirá una serie de palabras clave y debéis diseñar de algún modo un método de agregación o presentación del conjunto de palabras clave de todo el corpus.

5) (**optativo**) Utilizad la librería **WordCloud** de Python o similares para generar una nube de palabras del corpus. El objetivo de este apartado es obtener una representación visual de los términos más relevantes del corpus extraído.

Cada paso de los descritos anteriormente debe estar detallado en el notebook a entregar y resuelto con código Python propio.

- **Entregables:**

  1) Guión python (.py)
  2) Python Notebook (.pynb)

Es fundamental que el Notebook sea autoexplicativo de 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-). 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 la documentación con la 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). 2.5 puntos se corresponden a la correcta realización de los apartados obligatorios -apartados de 1) a 4)- y el 0.5 se corresponde con la correcta realización del apartado optativo.
Fecha límite entrega: **18 de octubre (antes de la clase de prácticas)**

Se permiten entregas retrasadas pero se reducirá la puntuación del siguiente modo:

- Cada día tarde reduce en un 10% la máxima nota alcanzable (es decir, cada día tarde resta un 0.3 puntos de la nota que se os asigne al valorar la práctica)