PRACTICA 1: Extracción de datos de una red social

- **Extracción de datos de una red social:**

En algunos casos, la fuente web (p.e. una red social) facilita la extracción de datos a través de APIs que facilitan el acceso a los contenidos. Es el caso de **Reddit**, una popular red social de código abierto. Trabajaremos con Reddit para construir una colección de datos, que contenga envíos (posts y/o comentarios) emitidos por usuarios en Reddit.

https://www.reddit.com/about/
https://www.reddit.com/wiki/index

Reddit API licensing guidelines: https://www.reddit.com/wiki/licensing

Reddit API Rules: https://github.com/reddit/reddit/wiki/API

Pasos a seguir:

1) cread un usuario en la red social Reddit con el cual accederemos a la API.
2) utilizad un "User-Agent" acorde a las instrucciones disponibles en Reddit API Rules.
3) existen distintos mecanismos para acceder a la API de Reddit y distintos wrappers. Aquí utilizaremos un wrapper para Python: PRAW: The Python Reddit Api Wrapper (https://praw.readthedocs.org/en/stable/). Familiarizaros con la documentación de PRAW. Tenéis un documento completo con la última documentación disponible aquí: https://media.readthedocs.org/pdf/praw/latest/praw.pdf.
4) el primer objetivo en esta práctica consiste en extraer el máximo número de posts y comentarios de una de las subcomunidades de Reddit (subreddits). Podéis escoger vosotros la comunidad con la que os interesa trabajar pero tiene que ser un subreddit con suficiente intensidad de posts y comentarios y con entradas textuales suficientemente voluminosas (no algo que simplemente consista en publicar enlaces). Una comunidad en la que se genere discusión y comentarios entre los usuarios es más interesante que una comunidad que sea meramente una secuencia de enlaces a lugares externos. Por ejemplo, las comunidades /r/Advice, /r/AskReddit o /r/nosleep son buenos ejemplos de subcomunidades con rico contenido textual e interacciones entre los usuarios. Utilizando las posibilidades del API de Reddit, definid al menos dos modos de extraer contenidos: a) los últimos contenidos (extrayendo el máximo número que permite Reddit) y b) los contenidos más populares (acorde a las valoraciones de Reddit).
5) el conjunto de entradas que obtengáis en el paso anterior, junto con sus comentarios asociados, deben ser almacenadas en disco en un formato adecuado. Para ello, definid un esquema XML que permita almacenar toda la información disponible (al menos, título, contenido, fecha, tipo de entrada -post o comentario-) y guardad toda la colección obtenida en un único archivo XML. Este archivo XML debe ser luego legible desde código Python utilizando, por ejemplo, la ElementTree XML API:
https://docs.python.org/2.7/library/xml.etree.elementtree.html
6) realizad un simple procesamiento del corpus anterior para vectorizar la colección y mostrar los términos con mayor ponderación tf/idf. Para ello:

a) instalad y familiarizaros con scikit-learn (http://scikit-learn.org/stable/) y, en particular, sus posibilidades para extraer características a partir de texto (sección 4.2.3 de la página http://scikit-learn.org/stable/modules/feature_extraction.html#feature-extraction) y el TfIdf Vectorizer:
http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

b) dado el corpus obtenido de Reddit, y considerando cada post o comentario como

un documento individual, utilizad el TfIdf Vectorizer (filtrando stopwords y todas aquellas palabras que aparezcan en menos de 10 docs) para vectorizar la colección y seguidamente mostrar los 10 términos más “centrales” en la colección. Entendiendo como más centrales aquellos cuya suma acumulada de tf/idf sobre todos los documentos es mayor.

Entregables:

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

- 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), que se dividen de la siguiente forma:
    - 2 puntos: programa python correcto y con las funcionalidades arriba indicadas
      - 0.75 puntos: extracción de datos
      - 0.5 puntos: creación XML
      - 0.75 puntos: vectorización de la colección
    - 1 punto: python notebook

Fecha entrega: 30 de Noviembre, a las 20h.

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

- Entrega entre 1 y 3 semanas tarde: la puntuación recibida será el 70% de la obtenida al corregir la práctica
- Entrega entre 3 y 5 semanas tarde: la puntuación recibida será el 60% de la obtenida al corregir la práctica
- Entrega más de 5 semanas tarde: la puntuación recibida será el 50% de la obtenida al corregir la práctica