

PRÁCTICA OBLIGATORIA EVALUABLE

CLUSTERING DE NOTICIAS

Objetivo

El objetivo es poner en práctica diferentes conceptos aprendidos durante el curso y relacionados con la Recuperación de Información y la Minería de Texto, utilizando las librerías de procesamiento de texto vistas en clase, para aplicarlo a una tarea real concreta.

Normativa de entrega

La entrega de esta práctica es **obligatoria** y puede realizarse **individualmente o por parejas**. A continuación, se detallan otros datos de interés relacionados con la normativa de entrega:

- La fecha límite de entrega de la práctica será el día **17 de marzo de 2019, a las 23:59h**.
- La entrega de la práctica se hará a través de Aula Virtual, empleando el **formulario** habilitado para ello (se habilitará más adelante, se pondrá una noticia en el foro de novedades cuando esté disponible).
- Se deberá subir un único archivo comprimido con todo el código fuente que se haya implementado y se necesite para ejecutar la práctica, así como una memoria explicando qué se ha hecho y por qué.

Enunciado

De toda la cantidad de información nueva disponible cada día en la Red, una buena parte se corresponde con noticias. Las noticias en Internet son una importante fuente de información, ya que permiten estar informados en cualquier momento y en cualquier lugar sin fronteras geográficas. Además, debido al fenómeno de la globalización, existe un interés mayor por conocer no sólo las noticias a nivel local o nacional, sino también a nivel internacional, por lo que cada día se consultan noticias escritas en otras lenguas directamente en los propios medios extranjeros. Así, los periódicos online y los portales de noticias se han convertido en una de las principales fuentes de información de actualidad.

Las noticias suelen relatar un hecho que tiene relevancia social, por lo que se trata de textos de actualidad (política, económica, deportiva, etc.), que no suelen ser muy extensos (depende de la temática de la noticia y del contexto actual de esa temática), con una temporalidad muy marcada y una estructura similar (por lo general título, sumario y cuerpo). Estas características influyen de manera directa en la forma de procesarlas, y de ellas se nutren los trabajos que se dedican a investigar diferentes formas de procesar los documentos de noticias para llevar a cabo distintas tareas. Con las noticias se realizan, entre otras tareas, detección y seguimiento de

eventos o sucesos; análisis de sentimientos y opiniones con respecto a mercados financieros, compañías, personas, etc.; generación de resúmenes; clasificación de noticias; predicción de los mercados mediante el análisis de noticias financieras, etc.

Una organización automática de las noticias donde éstas se agrupen por temática puede resultar de interés, ya que facilita el seguimiento de noticias de un tema determinado, evitando así el importante esfuerzo humano requerido para ello. El agrupamiento automático es un proceso de aprendizaje no supervisado cuyo fin es organizar una colección de objetos en diferentes grupos o clusters, de tal manera que los objetos del mismo cluster sean lo más similares posibles entre sí, manteniendo a su vez el menor grado de similitud posible en relación a los objetos pertenecientes a otros clusters.

En esta práctica se pide agrupar de forma automática noticias en base al tema o evento que describen. Las noticias provienen de diferentes fuentes y en dos idiomas distintos (español e inglés). Los pasos mínimos que se pueden dar para resolver un problema de clustering de noticias serían los siguientes:

1. Recopilación de noticias de diferentes fuentes mediante crawling.
2. Procesamiento de los documentos descargados para convertirlos en texto. Habitualmente los documentos descargados serán páginas web en formato HTML, por lo que mediante alguna librería se procesarán y se convertirán en texto.
3. Procesar el texto para generar una representación común de cada documento, por ejemplo, una representación vectorial donde cada vector representa un documento. Cada componente del vector representa un rasgo del vocabulario del problema (todos aquellos rasgos únicos de todos los documentos que intervienen en el problema) y tiene un peso concreto, que mide la importancia de ese rasgo en el documento.
4. A partir de una matriz de vectores de documentos se utilizaría un algoritmo de clustering que agruparía los documentos en grupos en base a la similitud de sus contenidos. En este problema cada grupo representa una temática diferente.

En esta práctica hay pasos del esquema anterior que no se deben llevar a cabo porque ya se dan hechos:

- Como colección de trabajo se dispone de varios documentos .html obtenidos de diferentes fuentes de noticias online, en inglés y español. Se trata de una colección pequeña, en total 22 documentos.
- Se proporciona un código fuente en Python (con la versión de Python 3.7) con un ejemplo completo de procesamiento de las páginas web en formato txt y agrupamiento mediante un algoritmo de clustering aglomerativo. En este caso se ha utilizado la librería de Machine Learning en Python *Scikit-Learn* (<http://scikit-learn.org/stable/>). Esta librería presenta diferentes algoritmos para análisis de datos y procesamiento de textos, y es muy común utilizarla junto con otras librerías de procesamiento de texto como NLTK y Spacy.

En la Figura 1 se presenta una de las funciones principales del código fuente que se proporciona.

```
def cluster_texts(texts, clustersNumber, distance):
    #Load the list of texts into a TextCollection object.
    collection = nltk.TextCollection(texts)
    print("Created a collection of", len(collection), "terms.")

    #get a list of unique terms
    unique_terms = list(set(collection))
    print("Unique terms found: ", len(unique_terms))

    ### And here we actually call the function and create our array of vectors.
    vectors = [numpy.array(TF(f,unique_terms, collection)) for f in texts]
    print("Vectors created.")

    # initialize the clusterer
    clusterer = AgglomerativeClustering(n_clusters=clustersNumber,
                                       linkage="average", affinity=distanceFunction)
    clusters = clusterer.fit_predict(vectors)

    return clusters
```

Figura 1. Agrupamiento de textos con un algoritmo de clustering aglomerativo.

Como se puede ver en la figura, de forma general el código lo que hace es un procesamiento muy básico del texto:

- 1) Mira cuáles son el conjunto de términos que aparecen en los textos de la colección, para así obtener el vocabulario con el que representar los documentos.
- 2) Representa cada documento mediante un vector donde cada término del vocabulario se pesa con la medida TF (frecuencia del término en el documento).
- 3) Agrupa con el algoritmo de clustering aglomerativo, al que le da como parámetros de entrada el número de grupos en los que tiene que agrupar los documentos (en la colección de trabajo las temáticas diferentes son cinco), el tipo de enlace (esto es particular del algoritmo de clustering utilizado) y la medida de similitud con la que comparar los vectores de los documentos.

En la Figura 2 se presenta la salida del código proporcionado. Como se puede ver la salida se corresponde con una lista de números (etiquetas), donde cada número representa uno de los clusters generados. Así, cada posición de la lista se corresponde con uno de los documentos de entrada (respetando el mismo orden en el que estaban en la lista de textos de entrada), de manera que aquellas posiciones de la lista con igual número se refieren a documentos que se han agrupado juntos.

```

C:\Anaconda\envs\Python3Env\python.exe C:/Users/Soto/MI_TRABAJO/DOCENCIA/URJC/Ma
['Adios Lindsey Vonn _ ElMundo.html',
'Al menos cinco muertos por un tiroteo en una vivienda de Texas.html',
'British Journalist to Be Still Alive.html',
'Esa vivo John Cantlie_.html',
'Gobierno británico John Cantlie vivo.html',
'Gobierno presiona salvar Presupuestos _ EL PAIS.html',
'John Cantlie believed to be alive - CNN.html',
'John Cantlie still alive.html',
'John Cantlie vivo.html',
'La era Trump_ Trump desafía al Congreso y promete en Texas _terminar_ el '
'muro con México - RTVE.es.html',
'Lindsey Vonn despedida_LaVanguardia.html',
'Lindsey Vonn icon - BBC Sport.html',
'Pedro Sanchez amenaza a los independentistas _ ElMundo.html',
'Pedro Sanchez baraja convocar las elecciones generales_LaVanguardia.html',
'Primer mitin de campaña de Trump será en El Paso, Texas _ CNN.html',
'Reino Unido Cantlie vivo.html',
'Sanchez elecciones generales el 14 de abril_LaRazon.html',
'Skiing icon Lindsey Vonn_Globalnews.ca.html',
'Texans oppose Trump border wall in Texas _ Fort Worth Star-Telegram.html',
'Trump blasts Beto O'Rourke, stands by wall at Texas rally.html',
'Trump offers socialism for the rich, capitalism for everyone else _ Robert '
'Reich _ Opinion _ The Guardian.html',
'War on Drugs is to Blame for Government Corruption in El Paso, Texas.html']
Prepared 22 documents...
They can be accessed using texts[0] - texts[21]
Created a collection of 20061 terms.
Unique terms found: 5621
Vectors created.
test: [0 0 1 0 0 0 2 1 0 0 0 1 0 0 0 0 1 1 1 1 3]
reference: [0, 4, 2, 2, 2, 3, 2, 2, 2, 1, 0, 0, 3, 3, 1, 2, 3, 0, 1, 1, 1, 1]
rand_score: -0.029689608636977085

Process finished with exit code 0

```

Figura 2. Resultado de la ejecución del código de partida.

¿Qué se pide con la realización de esta práctica?

Si se analiza el código proporcionado detenidamente se podrá comprobar que el procesamiento que se ha hecho de los textos de las páginas web de las noticias es muy simple. Por ejemplo, no se están eliminando palabras vacías, no se está haciendo stemming ni lematización, no se reconocen Entidades Nombradas, etc. Esto no es lo habitual en este tipo de problemas. Por lo tanto, lo que debe hacer el alumno es una labor de investigación, de forma que vaya modificando el código, añadiendo nuevo procesamiento con el objetivo de generar la representación de los documentos más adecuada para el problema que se quiere resolver.

Con la métrica empleada para evaluar el resultado de clustering (*Adjusted Rand Index*, ARI^1) se podrá ver si cada modificación que se haga mejora o empeora los resultados obtenidos. Este índice indica un **mejor resultado cuanto más próximo a 1 sea**.

¹ https://en.wikipedia.org/wiki/Rand_index

En los trabajos de investigación dedicados a la tarea de clustering de noticias se han utilizado diferentes enfoques, utilizando datos de entrenamiento o no, utilizando diferentes algoritmos, etc. En esta práctica no se utilizarán datos de entrenamiento y el algoritmo de clustering puede ser el propuesto en el código de partida sin necesidad de cambiar a otro. Con respecto a cómo representan las páginas web los diferentes trabajos de investigación sobre esta tarea, también hay mucha variedad: algunos sólo utilizan Entidades Nombradas, o bien las utilizan de forma diferenciada, o utilizan n-gramas, o sólo utilizan parte del documento teniendo en cuenta la estructura de las noticias, etc.

El objetivo de la práctica no es generar una representación de los documentos muy compleja, sino hacer cambios sobre el código proporcionado con cierto criterio, con el fin de mejorar los resultados de agrupamiento. Para ello se tendrá que investigar un poco acerca de esta tarea de clustering de noticias para ver qué se suele hacer, investigar un poco el contenido de las páginas web para tratar de ver qué puede ser más importante a la hora de representar las páginas y también investigar cómo utilizar las librerías de procesamiento de texto vistas para hacer las modificaciones necesarias en la representación de los documentos.

Es importante tener en cuenta que a veces un cambio puede no producir mejoras, pero acompañado de otro cambio adicional sí supone una mejora.

Por último, además de modificar y extender el código que se proporciona, **se deben explicar qué cambios se han introducido y por qué**. Es muy importante que se vayan describiendo los diferentes cambios realizados, junto con un análisis de si funciona mejor o peor y **las razones por las que se cree que ocurre**. Hay que justificar cada paso que se dé y analizar si ha llevado a mejoras o no y porqué puede ser.

Pistas

1. Entidades Nombradas

El estilo de redacción de las noticias se corresponde con el género informativo, donde lo primordial es informar. Los elementos que debe reunir una noticia se conocen en el argot periodístico con el nombre de las 6W” (*What, Who, When, Where, Why, How*): ¿Qué sucedió? ¿A quién le sucedió? ¿Cuándo sucedió? ¿Dónde sucedió? ¿Por qué sucedió? y ¿Cómo sucedió? Estos seis elementos no son necesarios en todas las noticias, algunos de ellos pueden faltar o se pueden unir. El orden en el que aparecerán las respuestas a estas preguntas dependerá del suceso que se relate, del redactor, o incluso de la guía de estilo del medio que publique la noticia. Las respuestas a las diferentes preguntas que se pueden plantear ante una noticia están presentes en el contenido de esta y varias de esas respuestas las pueden proporcionar las Entidades Nombradas, como se puede apreciar en la Figura 3, donde están resaltadas con diferentes colores las ENs de distinta categoría.

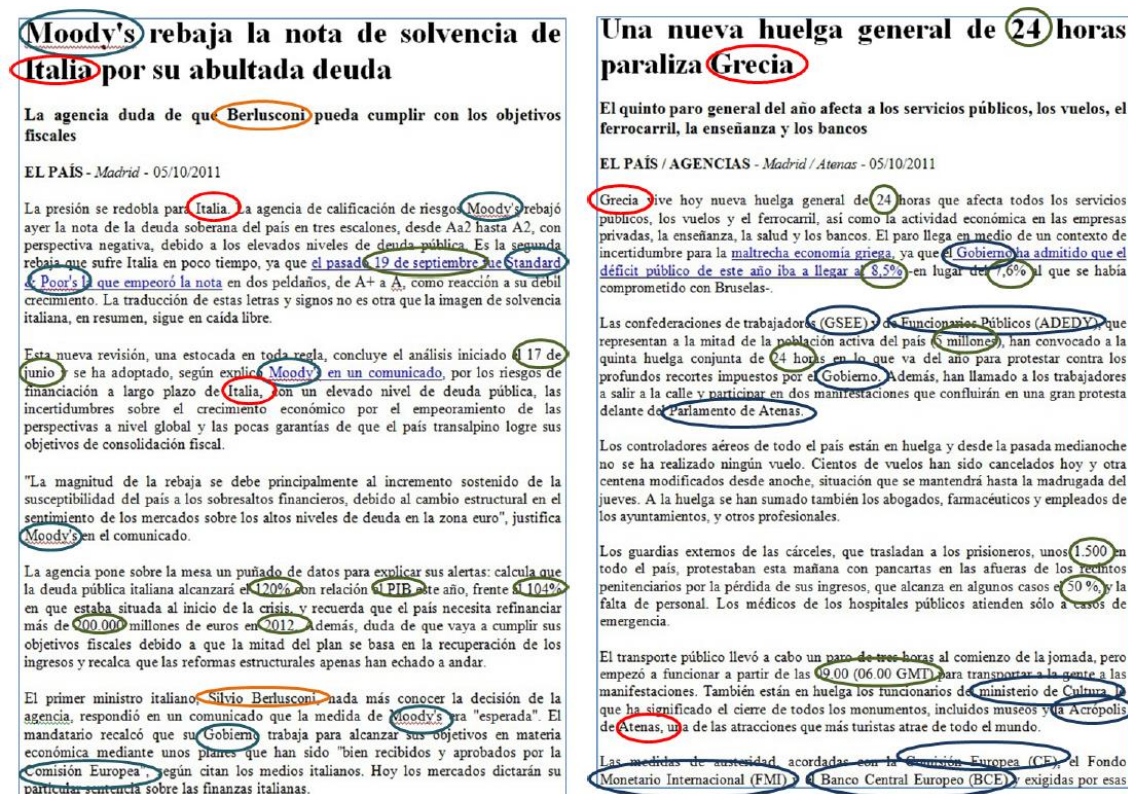


Figura 3. Ejemplo de noticias con las ENs resaltadas.

Es habitual que las entidades contengan información acerca de los protagonistas del suceso o evento que se describe en la noticia, y también cuándo y dónde se produce. Se trata de unidades de texto muy informativas, especialmente en el caso particular de textos de noticias. Por ello, se puede pensar en tenerlas en cuenta de forma diferente al resto de palabras del contenido de las noticias.

2. Posibilidad de traducción

Cuando se manejan textos en varios idiomas puede ser necesario recurrir a herramientas de traducción automática para unificar el lenguaje y que todos los textos estén en un único idioma. O para traducir sólo las partes de texto que interesen entre los idiomas que interesen o cualquier otra opción de traducción posible.

Existen diferentes APIs para traducir en Python, una de las más conocidas es *Goslate: Free Google Translate API*², que permite realizar traducciones utilizando el servicio web de traducción de Google, de forma rápida, sencilla y gratuita. Sin embargo, Google ha actualizado su servicio de traducción para prevenir que programas de crawling o similares puedan acceder a su servicio. De esta forma Goslate no se ha actualizado para "saltarse" este mecanismo de prevención de Google. Por ello, como alternativa se puede utilizar la

librería TextBlob³, que permite procesar textos en Python, pudiendo realizar diferentes tareas, entre ellas la traducción automática.

¿Cómo usarlo?

1. Instalación

Con el entorno virtual activo en el que se desee instalar, ejecutar

```
pip install textblob
```

2. Uso.

TextBlob permite no sólo traducir, sino también detectar el idioma de un texto. A continuación, se presenta un ejemplo:

```
from textblob import TextBlob

t = TextBlob("Hello World")

print('original language: ', t.detect_language())
print('Spanish translation: ', t.translate(to='es'))

print('French translation: ', t.translate(to='fr'))
```

Requisitos mínimos que debe tener la propuesta presentada

Además de estar bien todos los pasos dados en el procesamiento de los textos bien explicados y justificados, analizando el resultado de cada paso, es obligatorio trabajar con Entidades Nombradas de alguna forma.

Requisitos mínimos que debe cumplir la memoria escrita

Esta sección del enunciado pretende aportar algunas directrices para la redacción de la memoria:

- 1) La memoria debe escribirse en español, con un mínimo de 15 páginas sin contar la portada y el índice en caso de que se incluya.
- 2) Debe tener un formato estructurado con secciones y subsecciones.
- 3) Las imágenes no deberían ser mayores de un tercio de una página.
- 4) El esqueleto debe seguir una estructura aproximada a:
 1. Portada
 2. Índice (opcional)
 3. Introducción
 4. Propuestas con sus experimentos y sus resultados
 5. Conclusiones

