Universidad Rey Juan Carlos

TEMA 2 – HOJA DE EJERCICIOS IV

Se plantean diferentes ejercicios para utilizar recursos lingüísticos para diferentes tareas.

1. Se quiere desarrollar un evaluador de sentimientos basado en el uso de un recurso lingüístico externo, en concreto, las listas de palabras de opinión que tiene NLTK:

```
from nltk.corpus import opinion_lexicon
print(opinion_lexicon.words())
print(opinion_lexicon.words()[10:20])
print(opinion_lexicon.positive())

print(opinion_lexicon.negative())
```

```
['2-faced', '2-faces', 'abnormal', 'abolish', ...]
['aborts', 'abrade', 'abrasive', 'abrupt', 'abruptly', 'abscond', 'absence', 'absent-minded', 'absentee', 'absurd']
['a+', 'abound', 'abounds', 'abundance', 'abundant', ...]
```

   Dadas las siguientes opiniones, hay que indicar si son positivas, negativas o neutras.

   Opinión 1: *"Visceral, stunning and relentless film making. Dicaprio's Herculean, almost purely physical performance and Hardy's wide-eyed intensity coupled with the almost overwhelming beauty of the landscape - those trees, the natural light, the sun peeking through the clouds, rendered the proceedings down to savage poetry. A hypnotic, beautiful, exhausting film."*

   Opinión 2: *"I saw this film on Friday. For the first 40 minutes involving spoken dialogue they need not have bothered. For me the dialogue was totally unintelligible with grunting, southern states drawl, and coarse accent that made it impossible to understand what they were saying."*

   Opinión 3: *"It was an idiotic film that produces a magnificent fascination."*

   Se puede desarrollar un sencillo algoritmo que busque los tokens del texto de entrada en las listas de palabras positivas o negativas, sumando una unidad si el token se encuentra en la lista de palabras positivas o restando una unidad si está en la lista de palabras negativas. Si la puntuación global es > 0 sería una opinión positiva, si es < 0 sería negativa y si es igual a 0 se consideraría neutra.

   Nota: comprobar si una palabra puede estar en ambas listas.

2. Se pide lo mismo que en el ejercicio anterior, pero en este caso utilizando SentiWordNet (disponible en NLTK). Esta base de datos proporciona valores positivos y negativos para ciertas palabras en un rango entre -1 y 1.

Se puede seguir la misma idea de algoritmo que en el caso anterior, pero hay que tener en cuenta que SentiWordNet nos proporciona puntuaciones para los diferentes sentidos que tiene una palabra. Se puede entonces considerar la puntuación de todos los sentidos de la misma palabra, restando a lo positivo la puntuación negativa. Puede ser interesante que la puntuación global se promedie de acuerdo con el número de sentidos.

Utilizar como entrada las mismas opiniones del ejercicio anterior. ¿El resultado es mejor o peor que el conseguido con el algoritmo del ejercicio 1?

2.1. Hacer una variante del ejercicio donde se tengan en cuenta primero la categoría gramatical del token para considerar únicamente los scores de los sentidos que coincidan con la categoría gramatical dada. ¿Ha mejorado el resultado o ha empeorado con respecto a versiones anteriores?

¿Tendría sentido aplicar primero WSD? Es decir, ¿consultar la polaridad según el sentido correcto de cada palabra en un contexto dado?

3. Se pide extender lo que se ha hecho en los ejercicios anteriores, pero ampliado para todas las opiniones contenidas en un fichero .csv (textsSentimentsPNN.csv), donde cada opinión está anotada con su polaridad (positiva, negativa o neutra). Un ejemplo del formato del archivo se muestra a continuación, en la primera columna estaría el texto de la opinión y en la segunda el sentimiento (Positive, Negative o Neutral).

```
Text,Sentiment
Enjoying a beautiful day at the park!,Positive
Traffic was terrible this morning.,Negative
Just finished an amazing workout! 💪,Positive
Excited about the upcoming weekend getaway!,Positive
Trying out a new recipe for dinner tonight.,Neutral
Feeling grateful for the little things in life.,Positive
Rainy days call for cozy blankets and hot cocoa.,Positive
The new movie release is a must-watch!,Positive
Political discussions heating up on the timeline.,Negative
Missing summer vibes and beach days.,Neutral
Just published a new blog post. Check it out!,Positive
```

Utilizando de nuevo SentiWordNet se pide predecir la polaridad de cada mensaje.

Una vez que se tenga la polaridad de cada mensaje, utilizar las métricas del paquete sklearn.metrics de la librería scikit-learn para obtener los valores de accuracy, precisión, recall y f-measure. En definitiva, se quiere poder cuantificar si se están clasificando bien las opiniones según su polaridad. Cuanto más cercano a 1 sea el valor que se obtiene con estas métricas, mejor estará clasificando cada mensaje.

Nota: Las mismas variaciones del ejercicio anterior considerando el PoS de cada token antes de consultar a SentiWordNet se pueden aplicar aquí.

En los siguientes ejercicios se trabajará con algunos de los corpus disponibles en NLTK. Se puede encontrar una descripción de estos aquí.

4. Utilizar el corpus de opiniones de películas disponible en NLTK (movie_reviews). Lo que se quiere es entrenar un clasificador (en este caso NaiveBayesClassifier) para que aprenda sobre opiniones de películas ya anotadas como positivas o negativas, y así, una vez entrenado, poder conocer la polaridad de nuevas opiniones que queramos probar.

   Se debe partir el conjunto de datos en entrenamiento y test, y entrenar un clasificador y evaluarlo en la parte de test.

   Lo que se pide realmente en el ejercicio es, una vez se tiene el clasificador entrenado, probarlo para que clasifique cada una de las tres opiniones que se han utilizado en los ejercicios 1 y 2 de este listado. Hay que comprobar si el resultado mejora o empeora.

   Se pueden hacer pruebas de todos los algoritmos propuestos con otras opiniones para comparar.

   Si se utiliza otro clasificador diferente, ¿podría variar el resultado? Probar a utilizar el clasificador de Máxima Entropía (maxent en NLTK).

5. Se quiere analizar la frecuencia de las palabras en un corpus. Para ello, se van a utilizar el corpus Gutenberg. Este corpus contiene una pequeña selección de textos del Proyecto Gutenberg (que contiene 25000 libros en formato electrónico). A continuación, se muestran los nombres de los ficheros que contiene el corpus.

```
import nltk
# Pintamos los nombres de ficheros del corpus gutenberg
print(nltk.corpus.gutenberg.fileids())
```

```
['austen-emma.txt', 'austen-persuasion.txt', 'austen-sense.txt', 'bible-kjv.txt', 'blake-poems.txt', 'bryant-stories.txt',
'burgess-busterbrown.txt', 'carroll-alice.txt', 'chesterton-ball.txt', 'chesterton-brown.txt', 'chesterton-thursday.txt', 'e
dgeworth-parents.txt', 'melville-moby_dick.txt', 'milton-paradise.txt', 'shakespeare-caesar.txt', 'shakespeare-hamlet.txt',
'shakespeare-macbeth.txt', 'whitman-leaves.txt']
```

   Se pide:

a. Mostrar la frecuencia de cada palabra en uno de los textos del corpus.
b. Visualizar las n palabras más frecuentes. Hacer una visualización en modo gráfico de barras.

6. Se quieren extraer los nombres propios contenidos en los textos del corpus treebank de NLTK. Este corpus contiene textos de noticias del Wall Street Journal y es un corpus anotado, tiene anotaciones a nivel de análisis sintáctico, como se puede ver a continuación:

```
from nltk.corpus import treebank

#print(treebank.fileids())

#print(treebank.words(''wsj_0045.mrg'))

#print(treebank.tagged_words(''wsj_0045.mrg''))

print(treebank.parsed_sents('wsj_0045.mrg')[0])
```

```
(S
  (SBAR-TMP
    (IN Since)
    (S
      (NP-SBJ (NN chalk))
      (ADVP-TMP (RB first))
      (VP (VBD touched) (NP (NN slate)))))
  (, ,)
  (NP-SBJ-1 (NN schoolchildren))
  (VP
    (VBP have)
    (VP
      (VBN wanted)
      (S
        (NP-SBJ (-NONE- *-1))
```

Se pide:

a. Extraer los nombres propios contenidos en uno de los textos del corpus.
b. Extraer los nombres propios contenidos en todo el corpus. Un ejemplo de salida donde se muestran los 50 nombres propios más frecuentes en el corpus (acompañados de su frecuencia de aparición) es lo siguiente:

```
[('U.S.', 197), ('New York', 73), ('Japan', 72), ('October', 61), ('Tuesday', 46), ('Congress', 45), ('September', 42), ('Tr
easury', 36), ('S&P', 35), ('Mrs. Yeargin', 34), ('Wall Street', 32), ('Columbia', 31), ('Chicago', 30), ('Big Board', 28),
('House', 25), ('New York Stock Exchange', 25), ('Inc.', 24), ('China', 24), ('Oct.', 24), ('Co.', 23), ('Nov.', 22), ('Lond
on', 22), ('Washington', 21), ('USX', 21), ('Mr. Hahn', 20), ('South Korea', 19), ('California', 19), ('March', 18), ('Toky
o', 18), ('Georgia Gulf', 18), ('Buick', 18), ('August', 17), ('Wednesday', 16), ('Dec.', 16), ('President Bush', 16), ('Cal
if.', 16), ('Campbell', 16), ('Sea', 16), ('Bush', 16), ('American Express', 16), ('Moody', 16), ('America', 15), ('Mr. Nixo
n', 15), ('Senate', 14), ('Cray Research', 14), ('June', 14), ('IRS', 14), ('San Francisco', 14), ('December', 14), ('U.K.',
14)]
```

Como se puede apreciar en el ejemplo de salida, cuando el nombre propio es compuesto (tiene más de una palabra) aparecen juntas, esto es lo que se debe de conseguir a la hora de mostrarlo en la salida.