TEMA 3 – HOJA DE EJERCICIOS I

Se plantean una serie de ejercicios relacionados con la representación de textos, algunos de ellos para ver qué representación es más eficaz para unas tareas determinadas.

1. Utilizando una representación de bolsa de palabras (BoW) se quiere que los rasgos del vocabulario no sean palabras, sino n-gramas.

   Los n-gramas son secuencias contiguas de n elementos en un texto, cruciales para captar la información contextual. Pueden mejorar la representación de los textos al no considerar palabras sueltas, sino combinaciones de estas, lo que proporciona un contexto más rico.

   Dadas las siguientes frases: “Estamos ya a finales de febrero.”, “En febrero sigue haciendo frío.”, “Esto es una frase de ejemplo que habla de un mes del año.” Se pide lo siguiente:

   a) Generar una representación BoW con bigramas (2-grams).
   b) Generar una representación BoW con unigramas, bigramas y trigramas.

   En cada caso se debe de imprimir el vocabulario generado y la matriz de rasgos-documentos.

   Nota: con `CountVectorizer` de `sklearn` se pueden especificar los n-gramas a la hora de crear los vectores de la bolsa de palabras.

2. Dadas varias frases, generar una representación BoW con función de pesado TF-IDF. Imprimir el vocabulario generado y la matriz de rasgos-documentos.

3. Dadas las siguientes frases:

   "El saque de Nadal es imparable.",
   "Me encanta jugar al tenis los fines de semana.",
   "¿Viste el último partido de Federer?",
   "Necesito una nueva raqueta de tenis.",
   "El torneo de Wimbledon es mi favorito.",
   "Practicar el revés es fundamental para mejorar.",
   "El tenis es un deporte que requiere mucha concentración.",
   "¿Quién es tu tenista favorito?",
   "La pista de tierra batida es muy exigente.",
   "Vamos a jugar un partido de dobles."

Se pide lo siguiente:

a. Generar sus unigramas y mostrar el vocabulario con la frecuencia de cada uno en toda la colección. Parte de la salida sería la siguiente:

```
al - 1
batida - 1
concentración - 1
de - 7
deporte - 1
dobles - 1
el - 5
encanta - 1
es - 6
exigente - 1
favorito - 2
federer - 1
fines - 1
fundamental - 1
imparable - 1
jugar - 2
la - 1
los - 1
me - 1
```

b. Utiliza pandas para mostrar los unigramas por documento (por frase). Parte de la salida sería la siguiente:

	al	batida	concentración	de	deporte	dobles	el	encanta	es	exigente
0	0	0	0	1	0	0	1	0	1	0
1	1	0	0	1	0	0	0	1	0	0
2	0	0	0	1	0	0	1	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	1	0	1	0
5	0	0	0	0	0	0	1	0	1	0
6	0	0	1	0	1	0	1	0	1	0
7	0	0	0	0	0	0	0	0	1	0
8	0	1	0	1	0	0	0	0	1	1
9	0	0	0	1	0	1	0	0	0	0

4. Dada una frase cualquiera, utilizando la librería NLTK, se pide los siguiente:
   a. Calcular y mostrar los bigramas
   b. Calcular y mostrar los trigramas.
   c. Haz lo mismo que en los dos casos anteriores, pero añadiendo relleno (inicio y fin de frase).

   El padding o relleno sirve para que cada componente del n-grama aparezca en todas las posiciones del n-grama. Por ejemplo, para la frase “Hoy es un día soleado por fin.”, los trigramas serían:

```
------------ trigramas

[('Hoy', 'es', 'un'),
 ('es', 'un', 'día'),
 ('un', 'día', 'soleado'),
 ('día', 'soleado', 'por'),
 ('soleado', 'por', 'fin.')]
```

   Como se puede ver, solo algunas palabras aparecen en todas las posiciones del n-grama. Al añadir relleno al principio (parámetro `pad_left=pad`) y al final

(parámetro `pad_right=pad`), se consiguen los siguientes trigramas para la misma frase:

```
[(None, None, 'Hoy'),
 (None, 'Hoy', 'es'),
 ('Hoy', 'es', 'un'),
 ('es', 'un', 'día'),
 ('un', 'día', 'soleado'),
 ('día', 'soleado', 'por'),
 ('soleado', 'por', 'fin.'),
 ('por', 'fin.', None),
 ('fin.', None, None)]
```

Ahora sí se cumple que todas las palabras están en las diferentes posiciones del trigrama.

5. Dadas las frases del ejercicio 3, se pide, utilizando la librería sklearn, obtener la matriz de similitud coseno.
    a. Deberás utilizar la función de pesado binaria.
    b. Deberás utilizar la función de pesado tf.
    c. Deberás utilizar la función de pesado tf-idf.

    Prueba a modificar las frases para hacer algunas más parecidas y ver el efecto que tiene cada uno de los pesados.

6. Utilizando el corpus movie_reviews de la librería NLTK, que es un corpus etiquetado con opiniones sobre películas, se pide utilizar el clasificador MultinomialNB para comprobar qué tipo de representación vectorial puede funcionar mejor para este problema. Hay que dividir el corpus en un 80% para entrenamiento y el 20% restante para test, de forma que para cada prueba que se haga hay que calcular el accuracy y ver cuál va mejor.
    a. Representar con BoW y pesado binario.
    b. Representar con BoW y pesado TF.
    c. Representar con BoW y pesado TF-IDF.
    d. Lo mismo que en los casos anteriores, pero haciendo diferentes preprocesamientos:
        i. En el preprocesamiento de los textos, eliminar las stopwords. Probar a obtener de nuevo el accuracy con los tres pesados anteriores.
        ii. Añadir a la eliminación de stopwords, la lematización, combinado con los tres pesados.

    ¿Se ha visto el efecto del preprocesamiento? ¿Hay alguna combinación que va peor que lo más básico?