Práctica de minería de texto

Vamos a hablar del módulo RE en python. Este tutorial esta sacado de la página web: «Dive into Python.» https://diveintopython3.net/regular-expressions.html.

Supongamos que queremos sustituir la palabra «ROAD» por la abreviatura por "RD." El código python podría ser así:

```
1 direccion="100 BROAD ROAD"
2 direccion.replace("ROAD", "RD.")
```

Como se puede ver, esto no es lo más conveniente, ya que se hacen sustituciones en casos donde no interesan. La solución es utilizar **expresiones regulares**. Las expresiones regulares son cadenas de texto que sirven para especificar patrones en texto. En python se implementan en el módulo re y también existen herramientas online como ayuda a la creación. La forma de definir una expresión regular es por medio de la **concatenación de carácteres y operadores**, de forma que se obtenga una expresión regular válida.

La validez de una expresión depende que siga las reglas gramaticales del lenguaje. Algunos caracteres especiales que incluyen las expresiones regulares con su significado:

- . : representa cualquier caracter menos el salto de linea.
- * : representa cero o más copias de la expresión regular precedente.
- + : representa una o más copias de la expresión regular precedente.
- ? : representa cero o una copia de la expresión regular precedente.
- [ ] : representa una clase de caracteres y representa un símbolo que este dentro de los corchetes.
- ^ : representa el inicio de la linea y también representa la negación dentro de los corchetes.
- $: representa el fin de una linea.
- | : busca coincidencias o con la expresión regular precedente o con la consecuente.

En python, además se incorporan las siguientes funcionalidades:

En python se incorporan las siguientes funcionalidades:

- \d : representa un dígito de 0 a 9.
- \D : representa cualquier caracter menos un dígito de 0 a 9.
- \w : representa cualquier caracter alfanumerico.
- \W : representa cualquier caracter menos los alfanumericos.
- \s : representa cualquier separador (espacios, tabuladores).
- \S : representa cualquier caracter menos los separadores.
- \b : representa al limite de una palabra.

1) *Rellene el fichero «ExpresionesRegulares.py» de forma que funcionen*

Se proporciona un archivo con una serie de valoraciones de películas. El formato de este archivo es el comentario y, separado por una coma, está la valoración. Una característica interesante del módulo RE es que, después de buscar, permite buscar en **grupos**, es decir, las subexpresiones que están entre paréntesis.

```
phonePattern = re.compile(r'^(\d{3})-(\d{3})-(\d{4})$')
phonePattern.search('800-555-1212').groups()
```

También hay un método para sustituir expresiones regulares por otras.

```
phonePattern.sub(r'\1.\2.\3', '800-555-1212')
```

Estos son los métodos más importantes del módulo re.

2) *Programe en código python un script en el fichero «script.py» que separe, utilizando expresiones regulares, cada una de las valoraciones y las escriba a diferentes ficheros de texto cuyo nombre sea la posición del texto seguido de positivo o negativo con extensión txt. Por ejemplo, la quinta valoración debería guardarse en un fichero con nombre* 5positivo.txt*. Elimine las etiquetas
 que están en los comentarios con expresiones regulares, teniendo en cuenta que puede haber espacios dentro de la etiqueta.*

Para aplicar los algoritmos más básicos de IR, vamos a utilizar el fichero IMDB_raiz.csv, en el que todas las palabras han sido «normalizadas». Es decir, todas las palabras han sido sustituidas por su raíz.

3) *Añada un método al script anterior para generar todos los ficheros con el fichero IMDB_raiz.csv y que se genere el diccionario con el índice invertido. Conteste a la pregunta cuantos documentos tienen las palabras «prison», «brutal» y no contienen «king»*

Recordando el método del modelo vectorial y utilizando los pesos dados por la siguiente fórmula:

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{si } tf_{t,d} > 0 \\ 0, & \text{en otro caso} \end{cases}$$

4) *Halle el número de la valoración con la puntuación de la consulta «humor oscar».*