05 – nltk – proyecto – predicción textual

1. Digramas. Lectura preliminar

En el libro de Data Science From Scratch (Joel Grus)

> http://math.ecnu.edu.cn/~lfzhou/seminar/[Joel_Grus]_Data_Science_from_Scratch_First_Princ.pdf

podrás encontrar un bonito capítulo sobre el procesamiento de lenguaje natural. El segundo apartado de este capítulo trata sobre modelos de *n-gramas*. Éste es tu punto de partida para realizar la práctica que te propongo seguidamente. Léelo antes de seguir, pues lo que se propone seguidamente se basa en este apartado.

2. Formación de *bigramas* a partir de un texto

Partimos de un texto de referencia, largo a ser posible, aunque nosotros (y tú mismo) durante el desarrollo de las funciones preliminares, usaremos un fragmento de texto pequeñito, tomado de Moby Dick, en la versión libre de la biblioteca Gutenberg:

> *Hace unos años —no importa cuánto hace exactamente—, teniendo poco o nin-gún dinero en el bolsillo, y nada en particular que me interesara en tierra, pensé que me iría a nave-gar un poco por ahí, para ver la parte acuática del mundo. Es un modo que tengo de echar fuera la melancolía y arreglar la circulación. Cada vez que me sorprendo poniendo una boca triste; cada vez que en mi alma hay un noviembre hú-medo y lloviznoso; cada vez que me encuentro parándome sin querer ante las tiendas de ataúdes; y, especialmente, cada vez que la hipocondría me domina de tal modo que hace falta un recio principio moral para impedirme salir a la calle con toda deliberación a derribar metódicamente el sombrero a los tran-seúntes, entonces, entiendo que es más que hora de hacerme a la mar tan pronto como pueda. Es mi sustitutivo de la pistola y la bala. Con floreo filosófico, Catón se arroja sobre su espada; yo, calladamente, me meto en el barco. No hay nada sorpren-dente en esto. Aunque no lo sepan, casi todos los hombres, en una o en otra ocasión, abrigan sentimientos muy parecidos a los míos respecto al océano.*

La primera tarea es descomponer el texto en tokens. Pongamos los primeros:

```
['hace', 'unos', 'años', 'no', 'importa', 'cuánto', 'hace',
'exactamente', 'teniendo', 'poco', 'dinero', 'en', 'el', 'bolsillo',
'nada', 'en', 'particular', 'que', 'me', 'interesara', 'en', 'tierra',
'pensé', 'que', 'me', 'iría', 'un', 'poco', ...
```

Y con esta lista, formar los pares de palabras consecutivas (bigramas):

```
[('hace', 'unos'), ('unos', 'años'), ('años', 'no'), ('no', 'importa'),
('importa', 'cuánto'), ('cuánto', 'hace'), ('hace', 'exactamente'),
('exactamente', 'teniendo'), ('teniendo', 'poco'), …
```

Si examinamos el texto, veremos que la palabra “hace” está seguida por “unos”, pero también en otro lugar por “exactamente” y, en otro, por “falta”. Otras palabras únicamente preceden a una, y otras, a varias, en ocasiones de forma repetida. Formamos un diccionario, *pal_siguientes*, que, a cada palabra, le hace corresponder la lista de las palabras que la siguen:

```
{'hace': ['unos', 'exactamente', 'falta'],
 'unos': ['años'],
 'que' : ['me', 'me', 'tengo', 'me', 'en',
          'me', 'la', 'hace', 'es', 'hora']
 ...}
```

Así,

```
print(pal_siguientes["hace"])
print(pal_siguientes["unos"])
print(pal_siguientes["que"])
```

```
['unos', 'exactamente', 'falta']
['años']
['me', 'me', 'tengo', 'me', 'en', 'me', 'la', 'hace', 'es', 'hora']
```

Deseamos ahora modificar el diccionario de forma que, para cada palabra, dé la lista de las siguientes, sin repetición, y en orden de mayor a menor frecuencia:

Por ejemplo, si la lista de palabras siguientes es la siguiente,

['me', 'me', 'tengo', 'me', 'en', 'me', 'en', 'hace', 'en', 'hace']

una función puede transformarla en la lista siguiente:

['me', 'en', 'hace', 'tengo']

Todas estas operaciones han de reunirse en una función.

3. Predicción de texto escrito

Ya sabemos formar un diccionario de palabras siguientes, de mayor a menor frecuencia. Ahora, deseamos diseñar una herramienta de ayuda a la escritura que funcione así: cuando escribimos, para cada palabra nuestra herramienta nos ofrece las posibles palabras que pueden continuar nuestra frase.

Hay que decir que conviene formar nuestro diccionario con un texto largo, y que las palabras han de añadirse en el mismo, frase a frase, porque la palabra final de una frase no tiene por qué continuar de ninguna manera predecible.

Ahora, pongamos a prueba nuestro diccionario, con otro libro distinto, así:

Formamos todos los bigramas de nuestro libreo de prueba, y cada uno de ellos es un test para nuestro diccionario. Buscamos en nuestro diccionario la primera palabra. Si la segunda es la primera que ofrece, anotamos un 1; si es la segunda, un 2, etc. Digamos que, si llegamos al 5º intento, o no está la palabra en el diccionario, la calificación de 5 es ya la peor posible, y lo dejamos ahí.

Calcula la eficiencia de nuestro diccionario con un par de textos, a ser posible con uno del mismo autor y con otro distinto, por si el estilo del autor tuviera alguna influencia en las secuencias de palabras empleadas.

Otro test posible es el siguiente: si la palabra más frecuente no es la continuación de la frase, damos una letra de la que deseamos escribir, y buscaos la más frecuente con esa letra (2), y así sucesivamente, nuevamente hasta 5.

4. WhatsApp

Como seguramente sabrás, es posible exportar tus WhatsApps a formato de texto. Hazlo, crea un diccionario con el texto de tus frases, digamos que hasta hace una semana, y ponlo luego a prueba con las frases de la última semana, que no han formado parte del inventario.