Funciones de Nltk python

Alumna: Lynda Sherlyn Garcia Pulido

Profesor: Oswaldo carrillo Zepeda

Asignatura: Arquitectura de Servicios

Actividad: Funciones

Grado: 6°

Grupo: D

Carrera: Ingeniería en Computación Inteligente

Fecha de entrega: 03-Mayo-2022

Funciones

Tokenizing

Conceptualmente es la división de texto o cadenas de texto en partes más pequeñas como frases, palabras o símbolos, utilizando así un concepto como el de «*divide y vencerás* «. El resultado de hacer el *Tokenizing* será una lista de *Tokens.*

Por ejemplo, si aplicamos el *Tokenizing* a un párrafo el resultado será una lista de *tokens* los cuales estarán compuestos de oraciones, si hacemos lo mismo, pero para una oración obtendremos una lista de *tokens* compuestos por las palabras que componen dicha oración.

Teniendo la siguiente oración podemos ver que al hacer el Tokenizing obtendremos dos tokens, el de color azul y el de color naranja. Estos a su vez puedes ser separados en palabras pero para el ejmeplo utilizaremos primero la separación por sentencias.

NLTK

NLTK es un módulo de Python que contiene muchas funciones diseñadas para su uso en el análisis lingüístico de documentos y en el procesamiento de lenguaje natural. Para poder utilizar las funciones de este módulo primero debemos importarlo con import.

download()

NLTK es un módulo muy grande, cuando lo descargan e instalan (si instalaron Anaconda, el paquete lo hizo por ustedes) no se descarga NLTK en su totalidad, también está modulado y las partes las pueden encontrar en línea. La función download() de nltk abrirá un pequeño navegador con el que pueden descargar módulos de NLTK.

punkt

Uno de los módulos que vamos a utilizar (y que conseguimos con la función download() es punkt. Éste módulo contiene modelos para la tokenización de textos.

word_tokenize()

Tras descargar el módulo punkt se le agrega a nltk la función word_tokenize. La función recibe como parámetro el texto que se quiere tokenizar, y como un segundo parámetro opcional el idioma (hay idiomas que tienen diferentes reglas de tokenización), y regresa una lista con todos los tokens del texto. Algo muy similar a lo que logramos con nuestro tokenizador de expresiones regulares.

set()

Vimos también la función set() que convierte una lista en un conjunto (un set. Los conjuntos tienen propiedades diferentes a las listas, NO están ordenados, pero sus elementos tampoco se pueden repetir. Para los ejemplos que usamos en la clase, esa propiedad nos fue de utilidad para eliminar todos los tokens repetidos

count()

Ésta es una función de las listas. Nos permite saber cuántos elementos con un mismo valor contiene la lista, el valor se recibe como parámetro de la función.

def()

Para definir funciones en Python utilizamos la instrucción def seguido del nombre de nuestra función y sus paréntesis, dentro de los paréntesis podemos poner el nombre de los parámetros que esperamos recibir y que se podrán usar en el cuerpo de la función como variables, por último ponemos dos puntos para comenzar el bloque te código que será el contenido de nuestra función.

Return()

Si queremos que nuestra función regrese un valor (al igual otras funciones que hemos visto como read()) es necesario que utilicemos la instrucción return seguido del valor que queremos regresar. Con esto podemos asignar el resultado de nuestra función a una variable.

Text()

Con ésta función de nltk podemos convertir una lista de tokens en un Text de NLTK. Este es un tipo de variable (como los hay listas, conjuntos, números, etc.), y como tal tiene funciones propias diseñadas para el análisis de textos y el procesamiento de lenguaje natural.

concordance()

Esta es una función del Text de NLTK. Nos permite obtener las concordancias de una palabra, es decir, la palabra en su contexto. La palabra se usa como parámetro de la función. Además, tiene los parámetros opcionales width= y lines= para cambiar la cantidad de caracteres que se toman de cada lado de la palabra y la cantidad máxima de resultados que muestra. Esta función es similar a print() y NO devuelve ningún valor.

similar()

Otra función del Text de NLTK. Nos permite obtener las palabras similares a una palabra que recibe como parámetro. Para encontrar dichas palabras similares, compara los contextos de las palabras y regresa la que tienen los contextos más parecidos.