Instituto Politécnico Nacional

Escuela Superior de computo

Preprocesamiento de tweets

Profesora: Dra. Vanessa Alejandra Camacho Vázquez.

Alumnos:

- Bernal Reséndiz Axel
- Hernández Ramírez Jaciel Isai
- Salazar Carreón Jeshua Jonatán
- Torres Abonce Luis Miguel

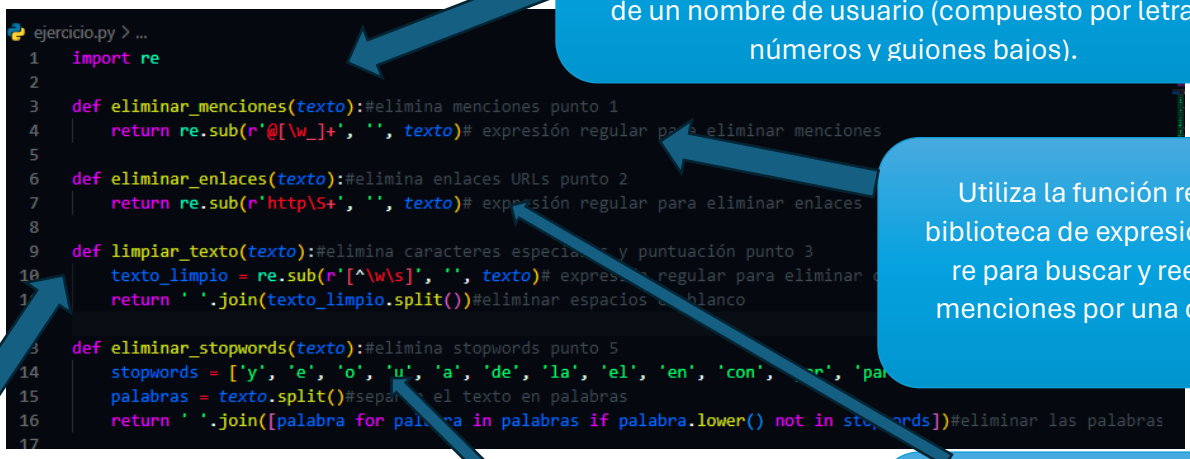
Grupo: 6CV1

Periodo escolar: 2023-2024

Contenido

Explicación

En el siguiente código hacemos el pre-procesamiento de tweets en español que nos implica una serie de tareas comunes para limpiar y estructurar los datos para poder facilitar su análisis posterior.



eliminar_menciones(texto):

Objetivo: Eliminar las menciones de usuarios en el texto, las cuales suelen comenzar con @ seguido de un nombre de usuario (compuesto por letras, números y guiones bajos).

Utiliza la función re.sub() de la biblioteca de expresiones regulares re para buscar y reemplazar las menciones por una cadena vacía.

limpiar_texto(texto):

Objetivo: Eliminar caracteres especiales y signos de puntuación del texto, y normalizar los espacios en blanco (por ejemplo, convertir múltiples espacios en uno solo).

eliminar_enlaces(texto):

Objetivo: Eliminar enlaces o URLs del texto.

eliminar_stopwords(texto):

Objetivo: Eliminar las palabras comunes que suelen aportar poco valor semántico al significado de un texto (conocidas como stopwords) para tareas de NLP.

Función procesar_tweet(tweet):

Esta función toma un único tweet como entrada y aplica una serie de transformaciones para limpiarlo:

```python
def procesar_tweet(tweet):
    tweet = tweet.lower()  # Convertir todo el texto a minúsculas punto 10
    tweet_sin_menciones = eliminar_menciones(tweet)
    tweet_sin_enlaces = eliminar_enlaces(tweet_sin_menciones)
    tweet_limpio = limpiar_texto(tweet_sin_enlaces)
    tweet_sin_stopwords = eliminar_stopwords(tweet_limpio)
    return tweet_sin_stopwords

def main():
    with open('tweets_asco.txt', 'r', encoding='utf-8') as arc
        tweets = archivo.readlines()#leer el archivo
    tweets_procesados = [procesar_tweet(tweet) for tweet in tw
    with open('tweets_asco_procesados.txt', 'w', encoding='utf
        for tweet in tweets_procesados:#escribir los tweets pr
            archivo_procesado.write(tweet + '\n')#escribir los


main()
```

Convierte el texto a minúsculas: Esto ayuda a estandarizar el texto, ya que, en el análisis de texto, las mayúsculas y minúsculas suelen considerarse iguales.

Elimina menciones: Utiliza la función eliminar_menciones para quitar las menciones de usuario que comienzan con @.

Elimina enlaces: Aplica eliminar_enlaces para quitar cualquier URL presente en el tweet.

Limpia el texto: Utiliza limpiar_texto para eliminar caracteres especiales, signos de puntuación y normalizar los espacios en blanco.

Elimina stopwords: Con eliminar_stopwords, filtra las palabras que suelen ser muy comunes y aportan poco valor semántico individual al análisis del texto.

Función main():

Abre y lee un archivo llamado 'tweets_asco.txt', que se asume contiene tweets sin procesar, uno por línea.

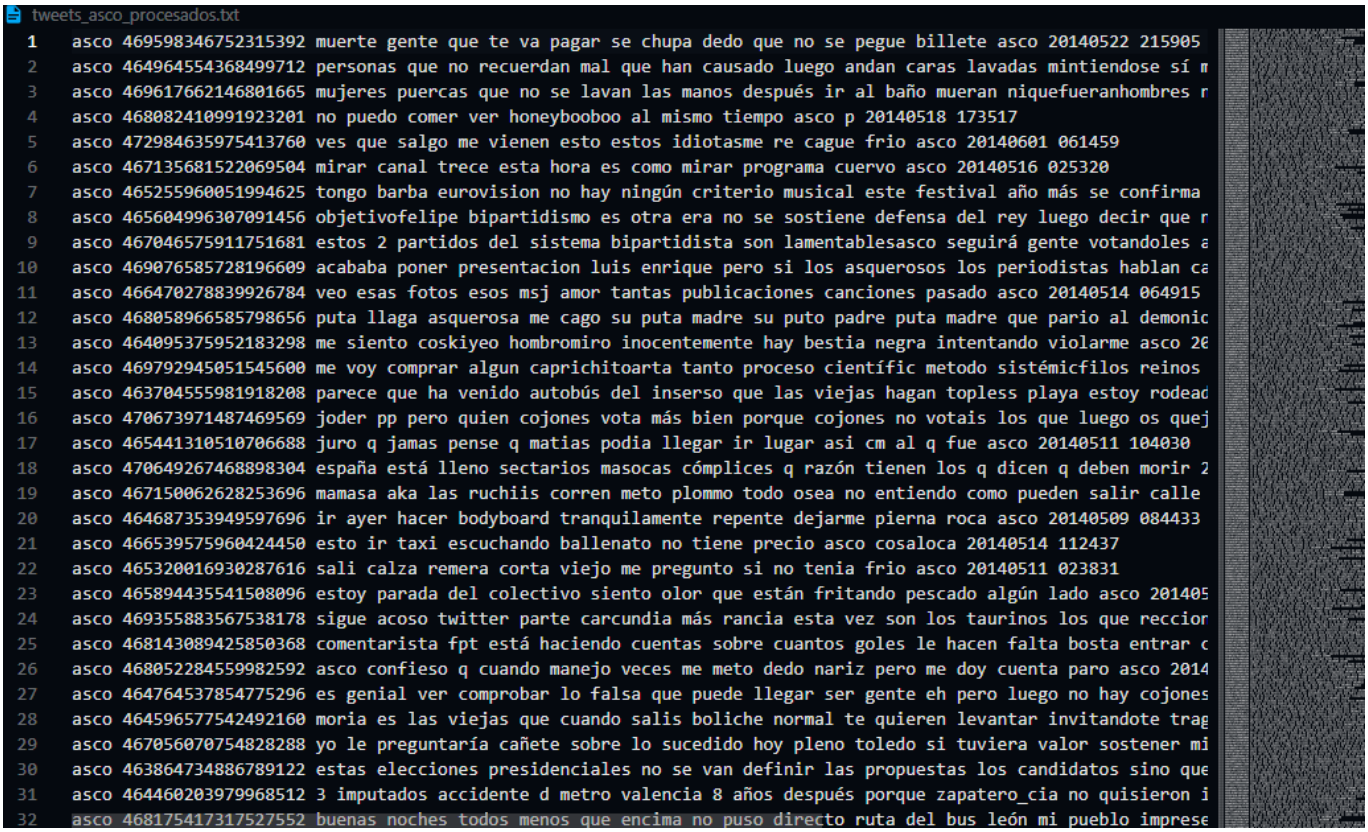
Procesa cada tweet leído del archivo utilizando la función procesar_tweet.

Escribe los tweets procesados en un nuevo archivo llamado 'tweets_asco_procesados.txt', donde cada tweet procesado se guarda en una nueva línea.

Esta función nos automatiza el flujo de trabajo desde la lectura del archivo de tweets crudos hasta la generación de un archivo con los tweets limpios y preparados para análisis.

Resultado:

Ahora se muestran los puntos 1, 2, 3, 5, 8 y 10, además tendrán que eliminar para cada tweet tanto su ID como su fecha y hora de creación.


tweets_asco_procesados.txt

```
1  asco 469598346752315392 muerte gente que te va pagar se chupa dedo que no se pegue billete asco 20140522 215905
2  asco 464964554368499712 personas que no recuerdan mal que han causado luego andan caras lavadas mintiendose sí m
3  asco 469617662146801665 mujeres puercas que no se lavan las manos después ir al baño mueran niquefueranhombres r
4  asco 468082410991923201 no puedo comer ver honeybooboo al mismo tiempo asco p 20140518 173517
5  asco 472984635975413760 ves que salgo me vienen esto estos idiotasme re cague frio asco 20140601 061459
6  asco 467135681522069504 mirar canal trece esta hora es como mirar programa cuervo asco 20140516 025320
7  asco 465255960051994625 tongo barba eurovision no hay ningún criterio musical este festival año más se confirma
8  asco 465604996307091456 objetivofelipe bipartidismo es otra era no se sostiene defensa del rey luego decir que r
9  asco 467046575911751681 estos 2 partidos del sistema bipartidista son lamentablesasco seguirá gente votandoles a
10 asco 469076585728196609 acababa poner presentacion luis enrique pero si los asquerosos los periodistas hablan ca
11 asco 466470278839926784 veo esas fotos esos msj amor tantas publicaciones canciones pasado asco 20140514 064915
12 asco 468058966585798656 puta llaga asquerosa me cago su puta madre su puto padre puta madre que pario al demonic
13 asco 464095375952183298 me siento coskiyeo hombromiro inocentemente hay bestia negra intentando violarme asco 20
14 asco 469792945051545600 me voy comprar algun caprichitoarta tanto proceso científic metodo sistémicfilos reinos
15 asco 463704555981918208 parece que ha venido autobús del inserso que las viejas hagan topless playa estoy rodead
16 asco 470673971487469569 joder pp pero quien cojones vota más bien porque cojones no votais los que luego os quej
17 asco 465441310510706688 juro q jamas pense q matias podia llegar ir lugar asi cm al q fue asco 20140511 104030
18 asco 470649267468898304 españa está lleno sectarios masocas cómplices q razón tienen los q dicen q deben morir 2
19 asco 467150062628253696 mamasa aka las ruchiis corren meto plommo todo osea no entiendo como pueden salir calle
20 asco 464687353949597696 ir ayer hacer bodyboard tranquilamente repente dejarme pierna roca asco 20140509 084433
21 asco 466539575960424450 esto ir taxi escuchando ballenato no tiene precio asco cosaloca 20140514 112437
22 asco 465320016930287616 sali calza remera corta viejo me pregunto si no tenia frio asco 20140511 023831
23 asco 465894435541508096 estoy parada del colectivo siento olor que están fritando pescado algún lado asco 201405
24 asco 469355883567538178 sigue acoso twitter parte carcundia más rancia esta vez son los taurinos los que reccior
25 asco 468143089425850368 comentarista fpt está haciendo cuentas sobre cuantos goles le hacen falta bosta entrar c
26 asco 468052284559982592 asco confieso q cuando manejo veces me meto dedo nariz pero me doy cuenta paro asco 2014
27 asco 464764537854775296 es genial ver comprobar lo falsa que puede llegar ser gente eh pero luego no hay cojones
28 asco 464596577542492160 moria es las viejas que cuando salis boliche normal te quieren levantar invitandote trag
29 asco 467056070754828288 yo le preguntaría cañete sobre lo sucedido hoy pleno toledo si tuviera valor sostener mi
30 asco 463864734886789122 estas elecciones presidenciales no se van definir las propuestas los candidatos sino que
31 asco 464460203979968512 3 imputados accidente d metro valencia 8 años después porque zapatero_cia no quisieron i
32 asco 468175417317527552 buenas noches todos menos que encima no puso directo ruta del bus león mi pueblo imprese
```

Información adicional

Import re

En Python se utiliza para importar el módulo re, que proporciona soporte completo para expresiones regulares, las expresiones regulares (RegEx) son una herramienta poderosa para el procesamiento de cadenas de texto, permitiendo la búsqueda, sustitución, y manipulación de texto basada en patrones definidos.

Aquí te detallo algunos de los usos más comunes del módulo re:

Búsqueda de Patrones: Puedes buscar si una cadena de texto contiene un patrón específico. Por ejemplo, verificar si un texto contiene direcciones de correo electrónico, números de teléfono, etc.

División de Cadenas: Permite dividir una cadena de texto en una lista, utilizando un patrón como delimitador, es más poderoso que el método. split() de las cadenas de texto estándar, ya que permite patrones complejos para la división.

Sustitución de Texto: Puedes reemplazar partes de una cadena que coincidan con un patrón dado por otro texto, esto es útil para limpiar o modificar textos, como eliminar enlaces o menciones en tweets, como has visto en el código anterior.

Extracción de Información: Es posible extraer partes específicas de un texto que coinciden con un patrón, esto es útil, por ejemplo, para extraer todos los enlaces de un documento HTML o las fechas en un formato específico dentro de un texto.

Algunas funciones comunes del módulo re incluyen:

- re.search(): Busca un patrón dentro de una cadena y devuelve un objeto de coincidencia si se encuentra el patrón.
- re.match(): Similar a re.search(), pero solo busca al principio de la cadena.
- re.findall(): Encuentra todas las coincidencias de un patrón dentro de una cadena y las devuelve como una lista.
- re.sub(): Sustituye las coincidencias de un patrón en una cadena por otro texto.
- re.split(): Divide una cadena por las ocurrencias de un patrón.

El módulo re utiliza una sintaxis especial para definir los patrones de búsqueda, que puede ser simple (como buscar palabras específicas) o compleja (como identificar direcciones de correo electrónico válidas), lo que lo hace extremadamente versátil y poderoso para el procesamiento de texto en Python.

Código Completo

```
import re
import nltk
from nltk import word_tokenize, pos_tag, ne_chunk
import re

def eliminar_menciones(texto):#elimina menciones punto 1
    return re.sub(r'@[\w_]+', '', texto)# expresión regular para eliminar
menciones

def eliminar_enlaces(texto):#elimina enlaces URLs punto 2
    return re.sub(r'http\S+', '', texto)# expresión regular para eliminar
enlaces

def identificar_emojis(texto):#Idenficar emojis
    emojis = {
    ":)": "emoji Cara feliz",
    ":(": "emoji Cara triste",
    ":D": "emoji Gran sonrisa",
    ":P": "emoji Sacando la lengua",
    ";)": "emoji Guiño",
    ":'(": "emoji Llorando",
    ":O": "emoji Sorprendido",
    ":/": "emoji Confundido o incierto",
    ":|": "emoji Neutral",
    ":*": "emoji Beso",
    "<3": "emoji Corazón",
    ":3": "emoji Sonrisa gatuna",
    "XD": "emoji Riendo mucho",
    ":')": "emoji Riendo con lágrimas",
    ":@": "emoji Enfadado",
    ":S": "emoji Confundido o perplejo",
    "O:)": "emoji Santo o inocente",
    "8-)": "emoji Usando gafas de sol",
    ":X": "emoji Sellado con un beso",
    ":": "emoji Sonrojado o avergonzado",
    ":#": "emoji Secretismo o silencio",
    ":&": "emoji Hablando",
    ":^)": "emoji Sonrisa nasal"
    }
    texto_modificado = texto
    for emoji, descripcion in emojis.items():
        texto_modificado = texto_modificado.replace(emoji,
'$'+descripcion+'$')
    return texto_modificado

def limpiar_texto(texto):#elimina caracteres especiales y puntuación
punto 3
    texto_limpio = re.sub(r'[^\w\s]$', '', texto)# expresión regular para
eliminar caracteres especiales y puntuación
    return ' '.join(texto_limpio.split())#eliminar espacios en blanco
```

```
def eliminar_stopwords(texto):#elimina stopwords punto 5
    stopwords = ['y', 'e', 'o', 'u', 'a', 'de', 'la', 'el', 'en', 'con',
'por', 'para', 'entre', 'un', 'una', 'uno', 'unas', 'unos']
    palabras = texto.split()#separar el texto en palabras
    return ' '.join([palabra for palabra in palabras if palabra.lower()
not in stopwords])#eliminar las palabras que estén en la lista de
stopwords

def extraer_entidades(texto):
    palabras_pos = pos_tag(word_tokenize(texto))
    entidades = ne_chunk(palabras_pos)

    entidades_extraccion = []
    for subtree in entidades:
        if isinstance(subtree, nltk.Tree):
            entidad = ' '.join([word for word, tag in subtree.leaves()])
            tipo_entidad = subtree.label()
            entidades_extraccion.append((entidad, tipo_entidad))

    return entidades_extraccion

def procesar_tweet(tweet):
    tweet = tweet.lower()  # Convertir todo el texto a minúsculas punto
10
    tweet_sin_menciones = eliminar_menciones(tweet)
    tweet_sin_enlaces = eliminar_enlaces(tweet_sin_menciones)
    tweet_emoji = identificar_emojis(tweet_sin_enlaces)
    tweet_limpio = limpiar_texto(tweet_emoji)
    tweet_sin_stopwords = eliminar_stopwords(tweet_limpio)

    return tweet_sin_stopwords

def main():
    with open('tweets_asco.txt', 'r', encoding='utf-8') as archivo:#abrir
el archivo
        tweets = archivo.readlines()#leer el archivo
    tweets_procesados = [procesar_tweet(tweet) for tweet in
tweets]#procesar los tweets
    with open('tweets_asco_procesados.txt', 'w', encoding='utf-8') as
archivo_procesado:#escribir los tweets procesados
        for tweet in tweets_procesados:#escribir los tweets procesados
            archivo_procesado.write(tweet + '\n')#escribir los tweets
procesados
main()
```