

# Analizador de texto

## Text parser

Autores: Juan Manuel Restrepo Urrego

Stiven Valencia Ramírez

IS&C, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: [juanmanuel.restrepo@utp.edu.co](mailto:juanmanuel.restrepo@utp.edu.co)

[stiven.valencial@utp.edu.co](mailto:stiven.valencial@utp.edu.co)

**Resumen—** Desarrollaremos y resolveremos un problema inicial que es tomar un texto y saber qué sentimientos, qué personajes y de qué tema principal trata el texto. Con solo darlo saber estos anteriormente mencionados y fue realizado con librerías de lenguaje natural, tomados de manera diferente para que sea más legible la lectura del código y separar los resultados obtenidos más fácil.

Recordar que durante el proceso de este analizador de texto primero probamos con oraciones, luego párrafos y por último un texto ya que es más fácil dividir, ver resultados, comparar y ver si son los resultados esperados, de lo que se tiene como objetivo final.

**Palabras clave—** analizador de texto, librerías, lenguaje natural, sentimientos, personajes, tema principal, código, resultados.

**Abstract- We will develop and solve an initial problem which is to take a text and know what feelings, what characters and what topic the text is about. By just giving it to know these previously mentioned and was done with natural language libraries, taken in a different way to make it more readable code reading and separating the results obtained easier.**

**Remember that during the process of this text analyzer we first tested with sentences, then paragraphs and finally a text because it is easier to divide, see results, compare and see if they are the expected results of what you have as a final goal.**

**Keywords---** text parser, libraries, natural language, sentiment, characters, main topic, code, results.

mostraremos con imágenes del código propio para dar evidencias.

Tomamos como entorno de desarrollo el editor y compilador online llamado Colab de google. como se puede prever está totalmente desarrollado en python.

Fue dividido y diseñado en 3 partes, sentimientos, personajes o nombres y el tema del texto. claro no hay problema con unirlos y que al pasar el texto muestre en una sola compilación los resultados de una manera pero ya que estamos dando el paso a paso verificando resultados será lo mejor para dar mejor explicación de estos.

## I.1 SENTIMIENTOS

Para empezar daremos solución al tema de sentimientos para esto le pasamos al analizador de texto un párrafo ya lo probamos con un texto completo pero para mostrar resultados se facilita con un párrafo como podemos ver esta dividido en 3 partes por 3 puntos seguidos entonces el analizador de texto de lenguaje natural los divide por estos puntos además de que con la librería Sentiment Intensity Analyzer, haber depurado tokenizado las palabras más importantes de este obtenemos resultados muy fáciles de entender y correctos estos son.

When you are truly interested in something, you never stop learning. I think this tastes awful, I don't like it.

Pequeño párrafo dado al analizador de texto para la prueba.

## I. INTRODUCCIÓN

Para dar inicio a este proyecto de analizador de texto, usaremos librerías de lenguaje natural tales como NLTK, también descargamos ciertas APIS de la librería propia como base de datos para entrenar el lenguaje natural mas adelante

```

I am sure that is the reason why education is so important.
neg : 0.0

neu : 0.659

pos : 0.341

compound : 0.567

When you are truly interested in something, you never stop learning.
neg : 0.0

neu : 0.517

pos : 0.483

compound : 0.7571

I think this taste awful, I dont like it.
neg : 0.505

neu : 0.495

pos : 0.0

compound : -0.6261

```

Figura 1. Resultados de sentimientos.

Como vemos el lenguaje natural separa los resultados en tres partes positivo, negativa y neutro. Por último da un compendio de lo que anteriormente dicho fue más positivo o negativo dando una generalización de todo el texto y en este caso fue negativo.

También entiende la parte negativa no solo como palabras erróneas si no que dependen demasiado del contexto y del conjunto de palabras, no solamente con pasarle algo con palabras negadas dice es negativo no, ya que como podemos ver puede entender la neutralidad bastante bien.

## I.2 NOMBRE Y PERSONAJES

En este caso fue más sencillo, primero y como en estos tres problemas primero depuramos el texto tokenizado lo realmente importante del texto, después colocamos tags en cada token diciéndonos que es cada palabra si es un verbo, pronombre, sujeto, etc. Por último tomamos los sujetos del texto, esto se hace de forma sencilla gracias al entrenamiento de la librería NLTK.

This isn't a very long sentence but it's full of interesting words, pablo.

Esta es la palabra en este caso usada para los resultados.

Pero antes de mostrar quien es el sujeto en el texto vemos cómo separamos primero el texto en las palabras más importantes y ya luego si se ve quien es el sujeto en el texto

Recordar que se hace con este trocito para ver que evidentemente se está haciendo de forma correcta ya que se probó con un texto sustancioso y lo hace de forma correcta, pero para ver a ojo si es correcta es más complicado.

```

[('This', 'DT'),
 ('ins't', 'VBZ'),
 ('a', 'DT'),
 ('very', 'RB'),
 ('long', 'JJ'),
 ('sentence', 'NN'),
 ('but', 'CC'),
 ('it', 'PRP'),
 ('s', 'VBZ'),
 ('full', 'JJ'),
 ('of', 'IN'),
 ('interesting', 'JJ'),
 ('words', 'NNS'),
 (',', ','),
 ('pablo', 'NN')]

```

```

[word for (word, tag) in tags if "NN" in tag]

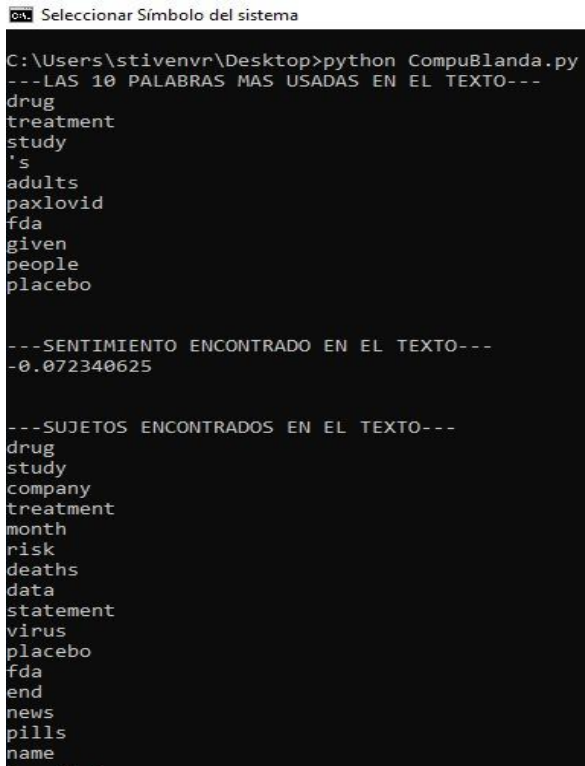
['sentence', 'words', 'pablo']

```

Figura 2. Resultados de nombre y personajes.

## I.3 TEMA PRINCIPAL

El tema principal fue el más complicado pero decidimos darnos a la tarea de dar las palabras claves del texto para que el usuario decida por sí mismo cuál es el tema principal del texto ya que anteriormente se debe tener una librería la cual inicializa el entrenamiento diciendo los temas, tuvimos muchas dificultades en este caso y decidimos solo dar las palabras más importantes del texto estos fueron los resultados.



```

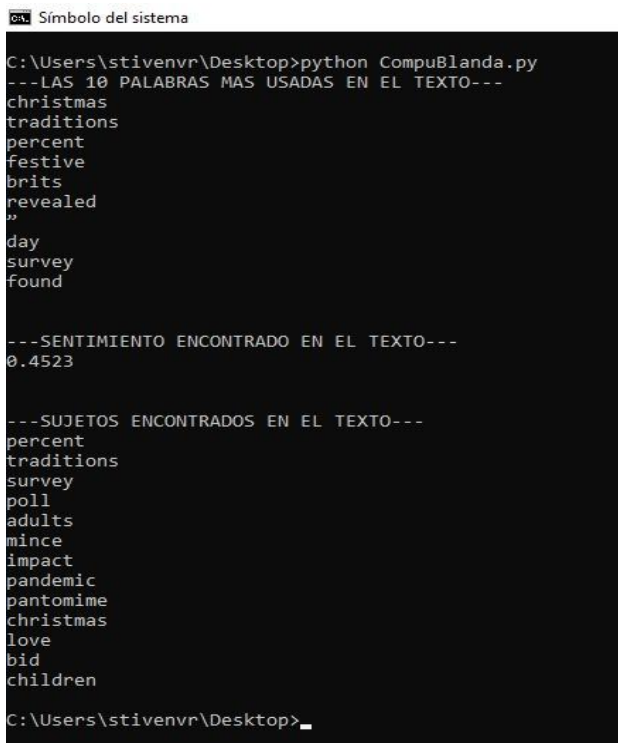
C:\Users\stivenvr\Desktop>python CompuBlanda.py
---LAS 10 PALABRAS MAS USADAS EN EL TEXTO---
drug
treatment
study
's
adults
paxlovid
fda
given
people
placebo

---SENTIMIENTO ENCONTRADO EN EL TEXTO---
-0.072340625

---SUJETOS ENCONTRADOS EN EL TEXTO---
drug
study
company
treatment
month
risk
deaths
data
statement
virus
placebo
fda
end
news
pills
name

```

Figura 3. Resultados totales del texto 1



```

C:\Users\stivenvr\Desktop>python CompuBlanda.py
---LAS 10 PALABRAS MAS USADAS EN EL TEXTO---
christmas
traditions
percent
festive
brits
revealed
"
day
survey
found

---SENTIMIENTO ENCONTRADO EN EL TEXTO---
0.4523

---SUJETOS ENCONTRADOS EN EL TEXTO---
percent
traditions
survey
poll
adults
mince
impact
pandemic
pantomime
christmas
love
bid
children
C:\Users\stivenvr\Desktop>

```

Figura 4. Resultados totales del texto 2

## II. ESTADO DEL ARTE

Aquí tenemos una gran cantidad de soluciones y código desarrollados y tomados para solucionar este problema, con solo mirar la librería nltk[1] de procesamiento de lenguaje natural ya podemos ver la cantidad de trabajo que hay detrás de este tema, pero si se debe saber dirigir este caso ya que otro tema importante es la depuración de información para el correcto análisis de lo que se quiere obtener para saber de verdad si un texto es positivo o negativo y neutro[2].

Para nombres y personaje se usó la misma librería pero diferente rama aquí la depuración es parecida pero se tokeniza diferente para saber que es cada palabra si es un verbo u cualquier otra cosa y tomamos solamente los sujetos o sea se hace dos depuraciones del texto una para palabras innecesarias y otra para cosas que realmente necesitamos en este caso los sujetos del texto[3].

## III. CONCLUSIONES

Lo más importante de todo el recorrido de este proyecto fue dividir el problema inicial en problemas más pequeños cuando ya se tenía en tres problemas diferentes se subdividen y se va escalando para obtener los resultados esperados.

Siempre se tuvo en cuenta en todo momento el escalamiento del problema como siempre se dijo primero una oración luego un párrafo y por último un texto así nos damos cuenta de la correcta depuración.

## REFERENCIAS

### Referencias en la Web:

- [1]  
<https://www.nltk.org/>
- [2]  
<https://www.arsys.es/blog/analisis-sentimientos-python-jupyter-notebooks/>
- [3]  
<https://www.instintoprogramador.com.mx/2019/07/resumen-de-texto-con-nltk-en-python.html>