
Proyecto 3 – Tecnologías Chapinas S.A.

202201444 – José David Góngora Olmedo

Resumen

La empresa Tecnologías Chapinas, S.A. está creando una herramienta para analizar el sentimiento de los usuarios en mensajes de Twitter. Esta herramienta lee mensajes que contienen referencias a otros usuarios (con el símbolo '@') y hashtags (con el símbolo '#'). Estos elementos pueden estar en cualquier lugar dentro del mensaje. Para determinar si un mensaje es positivo, negativo o neutro, la empresa utiliza un diccionario de palabras asociadas con sentimientos. Si las palabras en un mensaje coinciden con las del diccionario o si el número de palabras positivas y negativas es igual, se considera neutral. El programa que están desarrollando debe almacenar la información en formato XML después de recibir "n" mensajes de Twitter. Debe clasificar cada mensaje como positivo, negativo o neutro y generar un archivo con esta información para consultas y reportes posteriores.

Palabras clave

XML, Django, Mensajes, Sentimientos, Flask.

Abstract

The company Tecnologías Chapinas, S.A. is creating a tool to analyze user sentiment in Twitter messages. This tool reads messages that contain references to other users (with the '@' symbol) and hashtags (with the '#' symbol). These elements can be anywhere within the message. To determine whether a message is positive, negative or neutral, the company uses a dictionary of words associated with sentiment. If the words in a message match those in the dictionary or if the number of positive and negative words is equal, it is considered neutral. The program they are developing must store the information in XML format after receiving "n" messages from Twitter. It should classify each message as positive, negative or neutral and generate a file with this information for later queries and reporting.

Keywords

XML, Django, Messages, Feelings, Flask.

Introducción

El presente documento proporciona una visión general del proyecto desarrollado utilizando las tecnologías Python, Flask y Django. Este proyecto se centra en la creación de una aplicación destinada al análisis de sentimientos en mensajes provenientes de la red social Twitter. La herramienta, desarrollada por Tecnologías Chapinas, S.A., se ha diseñado para interpretar mensajes de usuarios de Twitter, identificando referencias a otros usuarios y hashtags en cada publicación.

La combinación de Python, un lenguaje versátil y potente, junto con los marcos de trabajo Flask y Django, ha permitido la creación de un sistema robusto y eficiente. Flask ha sido empleado para la construcción de servicios capaces de leer e interpretar los mensajes de Twitter, mientras que Django ha respaldado el desarrollo de un modelo de datos integral y la implementación de un mecanismo de almacenamiento y clasificación de información.

El objetivo principal de este proyecto es procesar mensajes de Twitter, aplicar un análisis de sentimientos basado en un diccionario definido por la empresa y generar informes a partir de esta clasificación. Este documento detallará la estructura, funcionalidades, componentes y procesos del proyecto, brindando una comprensión completa de la arquitectura y el funcionamiento interno de la aplicación.

Desarrollo del tema

Este programa fue diseñado para el manejo y obtención de mensajes a través de archivos con el formato XML.

Archivos de entrada

Principalmente son dos archivos de entrada uno con el diccionario de sentimientos positivos y negativos, otro con los mensajes el cual tiene fecha y el texto.

```
<?xml version="1.0"?>
<diccionario>
  <sentimientos_positivos>
    <palabra>bueno</palabra>
    <palabra>excelente</palabra>
    <palabra>cool</palabra>
  </sentimientos_positivos>
  <sentimientos_negativos>
    <palabra>malo</palabra>
    <palabra>pésivo</palabra>
    <palabra>triste</palabra>
  </sentimientos_negativos>
</diccionario>
```

Figure 1 Ejemplo del formato para el archivo diccionario

Y el otro archivo de entrada.

```
<?xml version="1.0"?>
<MENSAJES>
  <MENSAJE>
    <FECHA>Guatemala, 15/01/2023 15:25 hrs.</FECHA>
    <TEXTO> Bienvenido a USAC @estudiante01 @estudiante02,
    es un gusto que seas parte de esta institución, #bienvenidaUSAC#
  </TEXTO>
</MENSAJE>
</MENSAJES>
```

Figure 2 Ejemplo de archivo de entrada de mensajes

El programa cuenta con la parte del servidor el cual es el backend y el frontend que es la interfaz gráfica que mira el usuario.

Frontend:

En la página principal se encuentran dos apartados para subir los archivos de entrada, lado izquierdo para subir el archivo con el diccionario de las palabras y el lado derecho para subir el archivo que contiene los mensajes.



Figure 3 Página principal

Al subir los archivos y ver la alerta de que el proceso fue realizado correctamente podremos ver en el menú que se encuentra en la parte superior las otras opciones, la siguiente es sobre las menciones la cual nos permite encontrar o ver cuantas menciones tuvo un usuario en los mensajes publicados en un rango de fecha.

El rango de fecha se puede seleccionar en la parte de las fechas en la vista y al dar click en el botón de buscar se cargarán todos los datos encontrados y se mostrara una gráfica tipo barras sobre la consulta.



Figure 4 Pagina para consultar menciones

La siguiente opción es para consultar los hashtags que se agregaron al texto de los mensajes en un rango

de fechas, así como las menciones, los datos encontrados en los hashtags se mostraran en pantalla y se podrá ver una gráfica de barras sobre la consulta realizada.

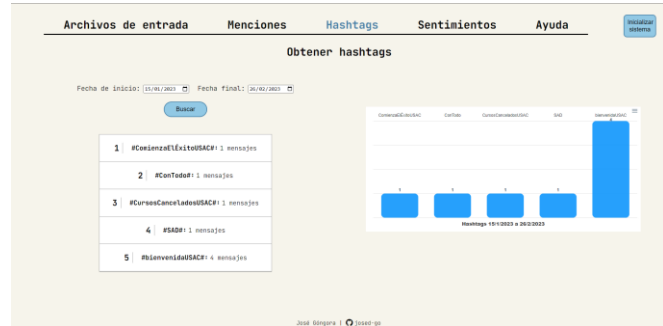


Figure 5 Pagina para consultar hashtags

La siguiente opción es sobre los sentimientos encontrados en los mensajes, esta opción analiza todos los mensajes y dependiendo que cantidad de palabras positivas y negativas tengo se denomina si es un mensaje positivo, negativo o neutro. Esto se basa en que, si la cantidad de palabras positivas en el mensaje es mayor a la cantidad de palabras negativas, este será un mensaje positivo, y viceversa con el tipo negativo. Si la cantidad de palabras positivas es igual a la cantidad de negativas el mensaje será denominado como tipo neutro.



Figure 6 Pagina para consultar los sentimientos

Y la última opción para navegar es la página de Ayuda en la cual se encuentran los datos del desarrollador y el link hacia esta documentación.



Figure 7 Pagina de ayuda

El botón que se encuentra en la parte superior derecha con el texto “Inicializar sistema” como su nombre lo indica permite al usuario inicializar el sistema y dejar todos los datos en 0, borra toda la información almacenada y nos permite empezar otra vez todo el proceso.

Archivos de salida

Al analizar los archivos XML de entrada, el programa es capaz de crear archivos XML de salida con la información resumida que encontró al momento de analizar los archivos de entrada. Lo que permite al usuario poder ver en formato XML lo que se analizó en los archivos.

resumenConfig.xml

Este archivo nos muestra la cantidad de palabras positivas y negativas que se encontraron en el archivo de entrada del diccionario, así mismo nos indica cuantas palabras positivas y negativas fueron rechazadas. Una palabra es rechazada cuando esa misma palabra ya se encontraba en el otro tipo.

```
<?xml version='1.0' encoding='UTF-8'?>
<CONFIG_RECIBIDA>
  <PALABRAS_POSITIVAS>24</PALABRAS_POSITIVAS>
  <PALABRAS_POSITIVAS_RECHAZADAS>1</PALABRAS_POSITIVAS_RECHAZADAS>
  <PALABRAS_NEGATIVAS_RECHAZADAS>1</PALABRAS_NEGATIVAS_RECHAZADAS>
  <PALABRAS_NEGATIVAS>24</PALABRAS_NEGATIVAS>
</CONFIG_RECIBIDA>
```

Figure 8 Ejemplo de archivo de salida resumenConfig

resumenMensajes.xml

Este archivo de salida nos muestra cuantos mensajes, cuantos usuarios y cuantos hashtags fueron encontrados para una fecha distinta.

```
<?xml version='1.0' encoding='UTF-8'?>
<MENSAJES_RECIBIDOS>
  <TIEMPO>
    <FECHA>15/01/2023</FECHA>
    <MSJ_RECIBIDOS>3</MSJ_RECIBIDOS>
    <USR_MENCIONADOS>6</USR_MENCIONADOS>
    <HASH_INCLUIDOS>3</HASH_INCLUIDOS>
  </TIEMPO>
  <TIEMPO>
    <FECHA>25/02/2023</FECHA>
    <MSJ_RECIBIDOS>2</MSJ_RECIBIDOS>
    <USR_MENCIONADOS>3</USR_MENCIONADOS>
    <HASH_INCLUIDOS>3</HASH_INCLUIDOS>
  </TIEMPO>
</MENSAJES_RECIBIDOS>
```

Figure 9 Ejemplo de archivo de salida resumenMensajes

Db_simulada.xml

Al no contar con base de datos como tal, este archivo muestra toda la información almacenada como los mensajes, palabras positivas, negativas, los usuarios y hashtags.

```
<?xml version='1.0' encoding='UTF-8'?>
<MENSAJES>
  <MENSAJE>
    <FECHA>15/01/2023</FECHA>
    <TEXTO> Bienvenido a USAC @estudiante01 @estudiante02,
    es un gusto que seas parte de esta institución, bueno, perfecto, lamentable
    #bienvenidaUSAC#
    </TEXTO>
    <USUARIOS_MENCIONADOS>
      <USUARIO>estudiante01</USUARIO>
      <USUARIO>estudiante02</USUARIO>
    </USUARIOS_MENCIONADOS>
    <HASHTAGS_MENCIONADOS>
      <HASHTAG>bienvenidaUSAC</HASHTAG>
    </HASHTAGS_MENCIONADOS>
    <TIPO_MENSAJE>Positivo</TIPO_MENSAJE>
    <CANT_PALABRAS_POSITIVAS>2</CANT_PALABRAS_POSITIVAS>
    <CANT_PALABRAS_NEGATIVAS>1</CANT_PALABRAS_NEGATIVAS>
  </MENSAJE>
</MENSAJES>
```

Figure 10 Ejemplo archivo de salida db_simulada

Conclusiones

- El programa tiene la capacidad para leer y escribir datos en archivo XML.
- El programa es capaz de recibir datos desde el frontend, mandar una solicitud al backend y devolver una respuesta.
- El programa es capaz de mostrar los datos de manera amigable al usuario y con información completa.
- La combinación de Flask para interpretar mensajes y Django para el modelo de datos ha permitido la creación de un sistema sólido y eficiente.
- El proyecto integra Python, Flask y Django para desarrollar una herramienta de análisis de sentimientos en mensajes de Twitter, lo que demuestra la versatilidad de estas tecnologías.

Referencias bibliográficas

- Welcome to Flask — Flask Documentation (3.0.X). (s. f.).
<https://flask.palletsprojects.com/en/3.0.x/>
- Django. (s. f.). Django Project.
<https://www.djangoproject.com/>
- ApexCharts. (2020, 7 marzo). ApexCharts.js - open source JavaScript charts for your website. ApexCharts.js. <https://apexcharts.com/>

Anexos

Para realizar las pruebas de las peticiones del backend se utilizó Postman.

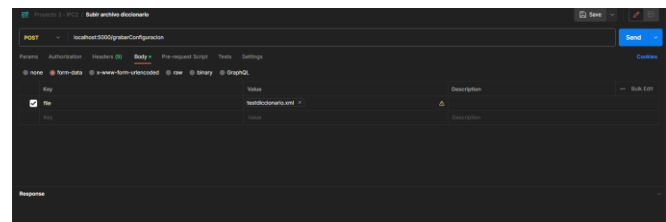


Figure 11 Petición para subir archivo de configuración

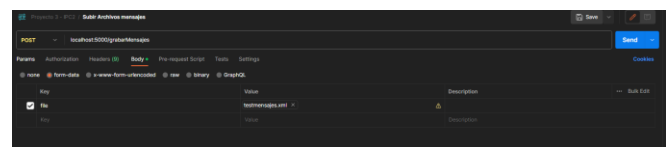


Figure 12 Petición para subir archivo de mensajes

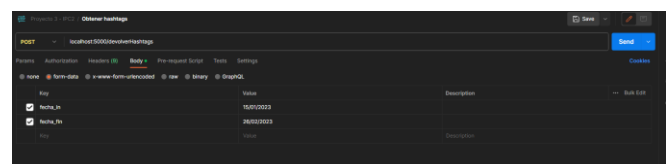


Figure 13 Petición para obtener las menciones

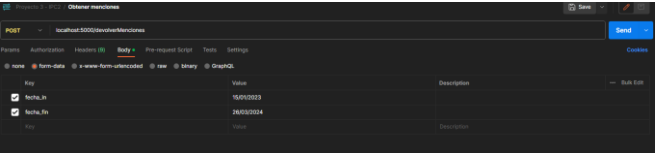


Figure 14 Petición para obtener los hashtags

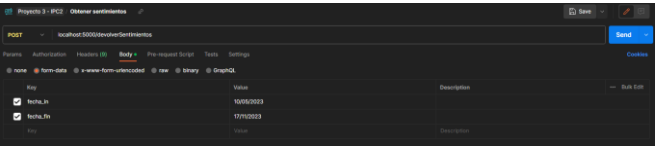


Figure 15 Petición para obtener los sentimientos

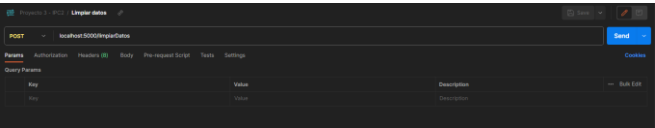


Figure 16 Petición para limpiar los datos

Diagrama de clases

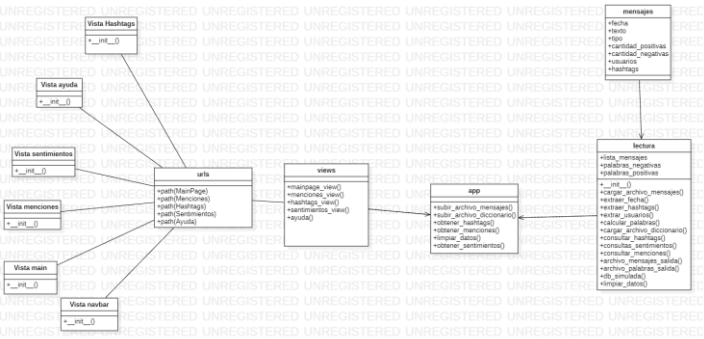


Figure 17 Diagrama de clases