
PROYECTO NUMERO 3

2022211515 – Gonzalo Fernando Pérez Cazún

Resumen

Como tal en nuestro proyecto numero 3 trabajaremos el frontend se centra en la interfaz de usuario y la experiencia del usuario, mientras que el backend se encarga de la funcionalidad y la gestión de datos detrás de escena en una aplicación o sitio web. Ambos aspectos son esenciales para el desarrollo de aplicaciones web completas y exitosas. El cual se comunicará con el backend es esencial para el funcionamiento interno de una aplicación o sistema, brindando la funcionalidad y el soporte necesarios para que el frontend pueda presentar la información de manera adecuada y permitir la interacción del usuario. Ambos, el frontend y el backend, son componentes fundamentales en el desarrollo de aplicaciones informáticas completas. Mientras para nuestra simular nuestra base de datos utilizaremos archivos xml el cual el backend extraer la información y se la manera a nuestro frontend según los solicitado.

Palabras clave

Frontend, Backend, xml y Framework

Abstract

In our project number 3, the frontend focuses on the user interface and user experience, while the backend is responsible for the functionality and data management behind the scenes in an application or website. Both aspects are essential for the development of successful and complete web applications. The communication between the frontend and the backend is crucial for the internal functioning of an application or system, providing the necessary functionality and support for the frontend to present information properly and enable user interaction. Both the frontend and the backend are fundamental components in the development of complete computer applications. For simulating our database, we will use XML files, from which the backend will extract the information and provide it to our frontend as requested.

Keywords

Frontend, Backend, xml and Framework.

Introducción

La empresa Tecnologías Chapinas, S.A. está desarrollando una herramienta que sea capaz de analizar contenido de redes sociales y establecer el sentimiento de los usuarios respecto a los mensajes emitidos en la red social.

Para lograr este fin, la empresa Tecnologías Chapinas, S.A. ha creado un servicio capaz de leer mensajes de Twitter que en esencia son textos que pueden contener 2 elementos

especiales:

- Una referencia a otro usuario de la red utilizando el símbolo @ y el nombre del usuario.
- Un texto entre símbolo # para definir un tema o hashtag para el mensaje

En un mensaje se puede hacer referencia a uno o más usuarios y colocar uno o más Hashtags en cualquier posición del texto.

Desarrollo del tema

El objetivo de nuestro programa La empresa Tecnologías Chapinas, S.A. ha creado una estrategia para establecer si un mensaje tiene un sentimiento positivo, negativo o neutro a través de la creación de un diccionario de datos que determine palabras que puedan calificar un mensaje como positivo o negativo, en caso de no tener palabras del diccionario de datos específico, o bien, que la cantidad de palabras con sentimientos positivos y negativos sean iguales, entonces, se considera que el mensaje es neutro.

El programa por desarrollar, luego de recibir “n” mensajes de la red social Twitter, deberá

almacenar la información necesaria en formato XML, para posteriormente emitir reportes y realizar consultas, además, debe considerar que cada mensaje solamente puede ser clasificado como positivo, negativo o neutro.

Cargar Palabras (Imagen 4):

En esta opción nosotros tenemos la opción de cargar las palabras que son positivas y las que son negativas al dar seleccionar se nos mostrara el como se mira nuestro archivo que cabe mencionar esta en formato de xml como se puede ver en la imagen 6, al pulsar el botón de enviar enviaremos nuestro archivo a nuestro backend el cual será una petición post, cuando nuestro backend revisa nuestro archivo lo primero que hará es evaluar si la el archivo de nuestra base de datos existe si no existe entonces la creara y formara la estructura de nuestro de nuestro archivo en donde se almacenaran nuestras palabras separadas. A como funciona nuestro algoritmo es que toma la raíz de nuestro archivo que le llega del fronted y luego recorre cada una de ellas de una vez va añadiendo a la base de datos pero antes debemos de evaluar si nuestra palabra si ya se encuentra en nuestra base de datos y si se encuentra en ella se le suma a una palabra como rechazada ya se si venia en positivos o negativos y si se encuentra se le sumara una a donde corresponde a positivos o negativas las palabras no tienen diferencia si son mayúsculas o minúsculas o si llevan tilde o no ,el cual nos servirá para poder devolver a nuestro fronted el cual debe de tener un formato como (Imagen 8).

Cargar Mensajes (Imagen 5):

En esta sección tenemos la opción de cargar los mensajes que tenemos los mensajes tendrán la estructura (Imagen 7), nosotros enviaremos una petición post a nuestro servidor el cual al recibirlo evaluara si existe nuestro archivo en donde esta nuestro base de datos sino esta entonces lo creara y la estructura para poder almacenar nuestros mensajes se nos solicita que debemos de retornar un xml como se ve en la (Imagen 9) el cual solicita el numero de menciones y hash recibidos en los mensajes separados por fecha pero la fecha viene en un formato en el cual trae información innecesaria para nosotros la fecha necesaria la abstraemos una gramática regular como esta:

`r'\d{2}/\d{2}/\d{4}'`

como antes mencionado para cada fecha abstraemos el número de menciones o hast distintos en dice fecha el cómo abstraemos el listado hast y menciones se explicara mas adelante pero lo que si almacenaremos en listas las veces que aparece ya sea una mención o hast las cuales cada una las cada lista crearemos un objeto el cual contenga la fecha y el listas antes mencionadas mas adelante las usaremos esto lo aremos junto estamos realizando la operación de guardar los mensajes en la base de datos que es nuestro archivo xml luego de terminar todo este proceso empezaremos a recuperar la información guardada en la lista de objetos como prosíguete recorreremos la lista linealmente guardando en un conjunto los elementos que ya operamos par ya no volver a repetir dichos elementos en dada interacción obtendremos las listas de menciones y hast y cada una la convertiremos en un objeto y de igual manera en el segundo for aremos lo mismo si las fechas son iguales

lo agregaremos al mismo conjunto y seguiremos hasta recorrer la lista desde los dos puntos esto con el fin de que cada conjunto de interacción crear un nuevo conjunto unión de ellos siguiendo la teoría de conjuntos. En la siguiente imagen se puede observar el código escrito en Python que opera nuestra descripción:

```
elementos_procesados = set()
fechasAgrupadas = []
for i in range(len(fechas)):
    fecha1 = fechas[i]

    if fecha1 not in elementos_procesados:
        fecha = []
        NumeroMensajes = 1
        fecha1 = fechas[i].fecha
        conjunto1_menciones = set(fechas[i].LISTA_USR_MENCIONADOS)
        conjunto1_hast = set(fechas[i].LISTA_HASH_INCLUIDOS)

        for j in range(i + 1, len(fechas)):
            fecha2 = fechas[j]

            if fecha2 not in elementos_procesados:
                fecha2 = fechas[j].fecha
                conjunto2_menciones = set(fechas[j].LISTA_USR_MENCIONADOS)
                conjunto2_hast = set(fechas[j].LISTA_HASH_INCLUIDOS)

                if fecha1 == fecha2:
                    conjunto1_menciones = conjunto1_menciones | conjunto2_menciones
                    conjunto1_hast = conjunto1_hast | conjunto2_hast
                    elementos_procesados.add(fecha2)
                    NumeroMensajes += 1

        listamenciones = list(conjunto1_menciones)
        listaconjuntos = list(conjunto1_hast)

        fecha.append(str(fecha1))
        fecha.append(str(NumeroMensajes))
        fecha.append(str(len(listamenciones)))
        fecha.append(str(len(listaconjuntos)))
        fechasAgrupadas.append(fecha)
        elementos_procesados.add(fecha1)
```

Al finalizar devolverá un xml que tiene el formato mostrado en la (Imagen 9)

Listar hast(Imagen 1)

En el fronted solicitara el rango de fechas en los cuales quiere consultar al darle el rango de menor a mayor entonces lo enviare a el backend esto lo hará a través de una petición get cuando lo obtenga el backend descomprimirá las fechas de inicio y fin en formato aaaa/mm/dd esto con el fin de poder evaluar de con una mayor facilidad la librería de *datetime* ya que esta maneja este formato luego de esto consultara si nuestra base de datos existe y si lo existiera retornaría un none, si encuentra la base de datos de mensajes entonces buscara la raíz, como queremos que los mensajes de una misma fecha se sumen todos hast entonces lo que haremos es que haremos una lista que tenga una fecha con referencia y en el una lista de hast. Luego de juntarlos contaremos cuantas veces

aparece cada hast y crearemos diccionarios los cuales si irán guardando en una lista la cual al finalizar el proceso retornaremos. Esto se mostrara a través del fronted en formade tablas.

Listar Menciones(Imagen 2)

En el fronted solicitara el rango de fechas en los cuales quiere consultar al darle el rango de menor a mayor entonces lo enviare a el backend esto lo hará a través de una petición get cuando lo obtenga el backend descomprimirá las fechas de inicio y fin en formato aaaa/mm/dd esto con el fin de poder evaluar de con una mayor facilidad la librería de *datetime* ya que esta maneja este formato luego de esto consultara si nuestra base de datos existe y si lo existiera retornaría un none, si encuentra la base de datos de mensajes entonces buscara la raíz, como queremos que los mensajes de una misma fecha se sumen todos las menciones entonces lo que haremos es que haremos una lista que tenga una fecha con referencia de fecha contenga todas las menciones. Luego de juntarlos contaremos cuantas veces aparece cada mension y crearemos diccionarios los cuales si irán guardando en una lista la cual al finalizar el proceso retornaremos. Esto se mostrara a través del fronted en formade tablas

Listar Sentimientos(Imagen 3)

En el fronted solicitara el rango de fechas en los cuales quiere consultar al darle el rango de menor a mayor entonces lo enviare a el backend esto lo hará a través de una petición get cuando lo obtenga el backend. La función divide la cadena fecha en sus

componentes de año, mes y día utilizando el método `split("-")`. Luego, crea objetos de fecha de Python utilizando la librería *datetime* a partir de estos componentes. Busca un archivo XML llamado `XML_Mensajes` utilizando la función `os.path.exists()`. Si el archivo no existe, la función devuelve `None`. Si si existe obtiene la raíz del archivo. La función crea una lista vacía llamada `fechas_en_rango` para almacenar las fechas que caen dentro del rango de fechas especificado por `fecha_inicio` y `fecha_fin`, recorre los elementos XML en `XML_Mensajes`, extrayendo la fecha y el texto de cada mensaje. Luego, compara la fecha de cada mensaje con el rango de fechas especificado. Si la fecha del mensaje está dentro del rango, se procesa el texto del mensaje, Para cada mensaje dentro del rango de fechas, se analizan las palabras del mensaje en busca de palabras positivas y negativas. Luego, se cuentan las palabras positivas y negativas en el mensaje, Se determina el tipo de sentimiento del mensaje (positivo, negativo o neutro) en función de la cantidad de palabras positivas y negativas, Para cada fecha en el rango que cumple con los criterios, se crea un objeto `Sentimiento` que contiene la fecha y el tipo de sentimiento. Estos objetos se agregan a la lista `fechas_en_rango`, se procede a procesar la lista de objetos `Sentimiento` para resumir y contar los sentimientos por fecha, se crea una lista llamada `result_list` que contendrá objetos `JSON` con información sobre el sentimiento para cada fecha luego se recorre la lista `fechas_en_rango`, contabilizando los sentimientos positivos, negativos y neutros para cada fecha. Luego, crea un objeto `JSON` con esta información y lo agrega a la lista `result_list` y Finalmente, la función devuelve la lista `result_list` en formato `JSON` utilizando la función `jsonify()`. Esta lista contiene información resumida

sobre los sentimientos para cada fecha en el rango especificado. Esto se mostrara a través del fronted en formade tablas-.

Ayuda

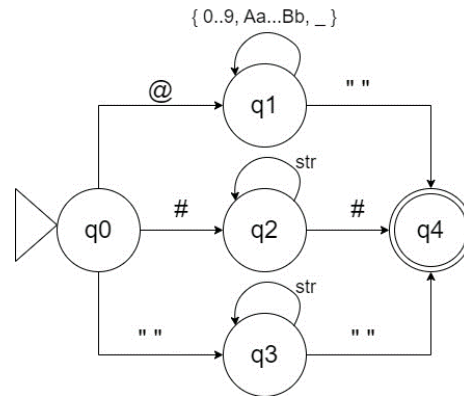
En esta sección mostraremos los datos del estudiante y a su vez este mismo manual.

Inicializar

Con esto nosotros eliminaremos todos los datos que se encuentran en nuestra bd de xml.

Algoritmo para que nuestras palabras no tengan distinción de si son mayúsculas, minúsculas o tildes, para empezar lo que se hace es convertir todas las palabras a minúsculas como el objetivo del software es el análisis de palabas no como están escritas entonces es osificante si están todas en mayúsculas o minúsculas y para evaluar las tildes lo que sea hace es enviarlas a una función donde tiene todas las igualdades esto con el fin que si la función nos devuelve un True es que son iguales, con esta descripción se puede describir como es que nuestro software evalúa esta parte.

Algoritmo para separa separar nuestro mensaje en palabras, mensajes y hast: Esto lo hice a través de un autómatas finitos no determinísticos como se muestra en la imagen:



En el estado de aceptación evalúa cual es su estado anterior y guarda el token en una lista que le corresponde.

Conclusiones

En este proyecto pudimos experimentar con las peticiones que le hace el frontend al backend tanto como el post el cual se utilizados para enviar datos desde un cliente a un servidor web. Y de que ambos pueden recibir tantos archivos con el formato de xml y json.

Referencias bibliográficas

HubSpot

<https://blog.hubspot.es/website/que-es-desarrollo-web>

Anexos

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Tabla de Hashtags

dd/mm/aaaa

dd/mm/aaaa

Buscar

Fecha	Hashtags	No. Repetidas
15-01-2023	#EstuSanCar@USC	2
15-01-2023	#USC	1
15-01-2023	#USC	1
16-01-2023	#USC	1

Imagen 1

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Tabla de Menciones

dd/mm/aaaa

dd/mm/aaaa

Buscar

Fecha	Menciones	no. Menciones
15-01-2023	@estudante01	2
15-01-2023	@estudante02	2
16-01-2023	@estudante01	1
16-01-2023	@estudante02	1

Imagen 2

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Tabla de Sentimientos en Mensajes

01/01/2023

01/01/2024

Buscar

Fecha	Mensajes con sentimiento positivo	Mensajes con sentimiento negativo	Mensajes neutros
15-01-2023	0	0	2
16-01-2023	0	0	1

Imagen 3

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Carga de Archivo XML De Palabras

Seleccione un archivo XML:

Seleccionar archivo

Un archivo seleccionado

Contenido del archivo XML:

Enviar

Resultado:

Imagen 4

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Carga de Archivo XML De Mensajes

Seleccione un archivo XML:

Seleccionar archivo

Un archivo seleccionado

Contenido del archivo XML:

Enviar

Resultado:

Imagen 5

PROYECTO 3

Lista Hashtags

Lista Menciones

Lista Sentimientos

Carga Palabras

Carga Mensajes

Ayuda

Inicializar

Ayuda

Nombre: Gonzalo Fernando Pérez Cazón

Carnet: 202211515

Cursos: Laboratorio Introduccion a la Programacion y Computación

Sección: A

Imagen 10

```
<?xml version="1.0"?>
<diccionario>
  <sentimientos_positivos>
    <palabra> bueno </palabra>
    <palabra> excelente </palabra>
    <palabra> cool </palabra>
    <palabra> satisfecho </palabra>
  ...
</sentimientos_positivos>
  <sentimientos_negativos>
    <palabra> malo </palabra>
    <palabra> pésimo </palabra>
    <palabra> triste </palabra>
    <palabra> molesto </palabra>
    <palabra> decepcionado </palabra>
    <palabra> enojo </palabra>
  ...
</sentimientos_negativos>
</diccionario>
```

Imagen 6

```
<?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>
```

Imagen 7

```
<?xml version="1.0"?>
<CONFIG_RECIBIDA>
  <PALABRAS_POSITIVAS> 4 </PALABRAS_POSITIVAS>
  <PALABRAS_POSITIVAS_RECHAZADA> 0 </PALABRAS_POSITIVAS_RECHAZADA>
  <PALABRAS_NEGATIVAS> 6 </PALABRAS_NEGATIVAS>
  <PALABRAS_NEGATIVAS_RECHAZADA> 0 </PALABRAS_NEGATIVAS_RECHAZADA>
</CONFIG_RECIBIDA>
```

Imagen 8

```
<?xml version="1.0"?>
<MENSAJES_RECIBIDOS>
  <TIEMPO>
    <FECHA> 15/01/2023 </FECHA>
    <MSJ_RECIBIDOS> 1 </MSJ_RECIBIDOS>
    <USR_MENCIONADOS> 2 </USR_MENCIONADOS>
    <HASH_INCLUIDOS> 1 </HASH_INCLUIDOS>
  </TIEMPO>
  ...
</MENSAJES_RECIBIDOS>
```

Imagen 9

Diagrama de clases

