
HERRAMIENTA QUE SEA CAPAZ DE ANALIZAR CONTENIDO DE REDES SOCIALES Y ESTABLECER EL SENTIMIENTO DE LOS USUARIOS

202300848 – Brayan Emanuel Garcia

Resumen

Se implementó una solución para clasificar mensajes según su sentimiento mediante un diccionario de palabras que identifica si un mensaje es positivo o negativo. Si un mensaje no contiene palabras relevantes del diccionario o si las palabras positivas y negativas se encuentran en igual cantidad, el mensaje se clasifica como neutro.

La arquitectura de la aplicación usa Flask para el backend y Django para el frontend, aprovechando sus capacidades para crear una interfaz de usuario intuitiva, que incluye funciones para generar reportes y analizar datos. Además, se integró JavaScript junto a Django para mejorar la accesibilidad de la interfaz de usuario, permitiendo una experiencia más dinámica. La aplicación también permite la carga de archivos XML, los cuales son procesados por el backend para mostrar la información resultante al usuario.

Palabras clave

Xml, flask, Django, Diccionario, Sentimiento

Abstract

A solution was implemented to classify messages by sentiment using a dictionary of words to determine if a message is positive or negative. If a message lacks relevant words from the dictionary or has an equal number of positive and negative words, it is classified as neutral.

The application's architecture uses Flask for the backend and Django for the frontend, leveraging their capabilities to create an intuitive user interface that includes reporting and data analysis functions. Additionally, JavaScript was integrated with Django to enhance user interface accessibility, providing a more dynamic experience. The application also allows for XML file uploads, which are processed by the backend to display the resulting information to the user.

Keywords

XML, flask, django, Dictionary, feelings.

Introducción

En el panorama digital actual, la capacidad de analizar y clasificar datos textuales es crucial para diversas aplicaciones, que van desde la retroalimentación de clientes hasta el análisis de sentimientos en redes sociales. Este proyecto tiene como objetivo desarrollar una solución web que aproveche un marco de análisis de sentimientos para evaluar mensajes como positivos, negativos o neutros. Mediante el uso de un diccionario de palabras cuidadosamente seleccionado, el sistema identifica el sentimiento transmitido en los mensajes enviados por los usuarios. El proyecto utiliza las fortalezas de Flask y Django, dos poderosos frameworks de desarrollo web en Python, para crear una experiencia de usuario fluida. Flask actúa como el backend, manejando la lógica y el procesamiento de datos, mientras que Django proporciona una interfaz frontend interactiva para los usuarios. Además, la integración de JavaScript mejora la accesibilidad y la capacidad de respuesta de la interfaz de usuario. La solución también incluye funcionalidades para procesar archivos XML, lo que permite a los usuarios subir y analizar sus datos de manera eficiente. Este proyecto no solo ilustra la aplicación práctica del análisis de sentimientos, sino que también destaca la importancia del diseño amigable con el usuario en el desarrollo de software.

Marco teorico

Para facilitar la comprensión del ensayo, a continuación ofrezco algunas definiciones de términos que podrían no ser tan comunes para personas que no están familiarizadas con el ámbito de la tecnología:

XML (eXtensible Markup Language): Es un formato de texto utilizado para almacenar y transportar datos de forma estructurada. A menudo se usa para compartir información entre diferentes sistemas o programas porque es fácil de leer para las

computadoras, pero también comprensible para los humanos. A diferencia de otros formatos de almacenamiento de datos, como bases de datos o archivos binarios, XML organiza los datos en una jerarquía mediante etiquetas.

Lista enlazada: Es una estructura de datos que se utiliza para almacenar una colección de elementos. A diferencia de un arreglo (o array), donde los elementos están almacenados en posiciones contiguas de memoria, en una lista enlazada cada elemento (o "nodo") contiene una referencia al siguiente, lo que permite añadir o eliminar elementos de forma dinámica y eficiente sin necesidad de reorganizar toda la estructura.

Simulación: En el contexto tecnológico, una simulación es un modelo computarizado que reproduce el comportamiento de un sistema real para estudiar su funcionamiento. En este proyecto, se simula el ensamblaje de productos en una línea de producción, permitiendo evaluar el rendimiento y optimizar el proceso sin tener que realizarlo físicamente.

Backend: Es la parte del software que no se ve ni interactúa directamente con el usuario, pero que realiza las operaciones fundamentales para que una aplicación funcione correctamente. El backend incluye bases de datos, servidores y la lógica que procesa las peticiones del usuario. En este caso, el backend del proyecto gestiona la carga de archivos, el procesamiento de instrucciones y la ejecución de simulaciones.

Interfaz gráfica (GUI, por sus siglas en inglés): Es el medio visual a través del cual los usuarios interactúan con un programa. Incluye botones, ventanas, menús y otros elementos gráficos que permiten al usuario utilizar el software de manera intuitiva, en lugar de tener que escribir comandos de texto.

Cola: En programación, una cola es una estructura de datos que sigue el principio de "primero en entrar, primero en salir" (FIFO por sus siglas en inglés). Imagina una fila de personas esperando su turno; la primera persona en la fila será la primera en ser atendida, y así sucesivamente. En este sistema, una cola se utiliza para gestionar las instrucciones que se van procesando durante la simulación.

Desarrollo del tema

Subtema 1: Lectura y Procesamiento de Archivos XML

El sistema inicia con la lectura de archivos XML que contienen empresas, servicios y mensajes. Para este propósito, se emplea la clase `app.py` que es llamada desde el `index.py`, que extrae los datos almacenados en los archivos y los convierte en estructuras manejables en Python, como listas enlazadas. La elección de listas enlazadas para el almacenamiento de matrices permite una gestión eficiente de grandes volúmenes de datos, ya que facilita la inserción y eliminación de elementos sin necesidad de reestructurar la memoria. Este enfoque es particularmente útil para la manipulación dinámica de matrices, optimizando el rendimiento general del sistema.

Subtema 2: Análisis de instrucciones

Cuando se crea la estructura de datos se pasan a una lista de caracteres y se almacenan las instrucciones en empresa y en mensaje. Esta se usa en `procesarInformacion` para poder buscar los sentimientos en el programa, también almacenamos los mensajes que están en el xml

Subtema 3: Generación de Graficos

Una de las características del programa es la muestra de gráficos renderizados directamente en la pagina a partir de los datos que se procesan del xml subido anteriormente.

Subtema 4: Simulación:

La simulación busca hacer matchear las palabras del diccionario con las de los mensajes para poder estudiar cuales son sus características y tipos de mensajes.

Subtema 5: Interfaz grafica:

La interfaz grafica hecha con html, css y un poco de Python para traer la información a tomar, empieza con pedirle al usuario que suba un archivo, luego de eso lo analiza, y nos permite la elección que tipo de informe queramos. Al seleccionarlal podemos presionar simulación, se llenara el espacio con un grafico que son de las instrucciones y con una tabla que describe de manera detallada el proceso optimizado, esto se guardara en un historial que podemos ver yéndonos a un el apartado de reportes.

Desde el apartado de reportes podemos hacer un informe de todos los mensajes y las Empresas, así como ver el historial de ensamblaje.

Subtema 6: Ayuda:

Al presionar o ir a esta parte nos mostrara el link del repositorio de la documentación y código fuente del programa, también la información del estudiante(autor) del programa y links para poder seguirlo en redes sociales y colocarse en contacto.

Conclusiones

El desarrollo de una herramienta capaz de analizar contenido de redes sociales y clasificar los sentimientos de los usuarios representa un avance significativo en el campo del procesamiento de datos textuales. La implementación de un sistema que utiliza un diccionario de palabras para identificar mensajes como positivos, negativos o neutros permite una comprensión más profunda de las opiniones y emociones expresadas en plataformas digitales.

La arquitectura del sistema, que combina Flask para el backend y Django para el frontend, garantiza una experiencia de usuario fluida y accesible. La integración de JavaScript en la interfaz gráfica no solo mejora la interactividad, sino que también facilita la visualización de datos, lo que es crucial para la toma de decisiones informadas.

El procesamiento de archivos XML, una característica clave del sistema, permite a los usuarios cargar y analizar datos de manera eficiente, asegurando que se puedan manejar grandes volúmenes de información sin comprometer el rendimiento. La capacidad de generar reportes y gráficos a partir de los datos procesados proporciona a los usuarios herramientas valiosas para interpretar los resultados de manera visual y comprensible.

Apéndices

Apéndice A: Ejemplo de Archivo XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<solicitud_clasificacion>
```

```
  <diccionario>
```

```
    <sentimientos_positivos>
```

```
      <palabra>excelente</palabra>
```

```
      <palabra>bueno</palabra>
```

```
      <palabra>genial</palabra>
```

```
      <palabra>agradable</palabra>
```

```
      <palabra>satisfactorio</palabra>
```

```
      <palabra>feliz</palabra>
```

```
    <palabra>contento</palabra>
```

```
  </sentimientos_positivos>
```

```
  <sentimientos_negativos>
```

```
    <palabra>malo</palabra>
```

```
    <palabra>horrible</palabra>
```

```
    <palabra>terrible</palabra>
```

```
    <palabra>insuficiente</palabra>
```

```
    <palabra>desagradable</palabra>
```

```
    <palabra>molesto</palabra>
```

```
    <palabra>decepcionado</palabra>
```

```
  </sentimientos_negativos>
```

```
<empresas_analizar>
```

```
  <empresa>
```

```
    <nombre>USAC</nombre>
```

```
    <servicios>
```

```
      <servicio nombre="inscripción">
```

```
        <alias>inscribir</alias>
```

```
        <alias>inscrito</alias>
```

```
        <alias>inscripción</alias>
```

```
      </servicio>
```

```
      <servicio nombre="asignación">
```

```
        <alias>asignar</alias>
```

```
        <alias>asignado</alias>
```

```
      </servicio>
```

```
      <servicio nombre="graduación">
```

```
        <alias>graduar</alias>
```

```
        <alias>graduación</alias>
```

```
      </servicio>
```

```
    </servicios>
```

```
  </empresa>
```

```
<empresa>
```

```
  <nombre>Banco Nacional</nombre>
```

```
  <servicios>
```

```
    <servicio nombre="cuenta">
```

```
      <alias>cuenta</alias>
```

```
      <alias>ahorro</alias>
```

```
    </servicio>
```

```
    <servicio nombre="tarjeta de crédito">
```

```
      <alias>tarjeta</alias>
```

```
      <alias>crédito</alias>
```

```
    </servicio>
```

```
    <servicio nombre="préstamo">
```

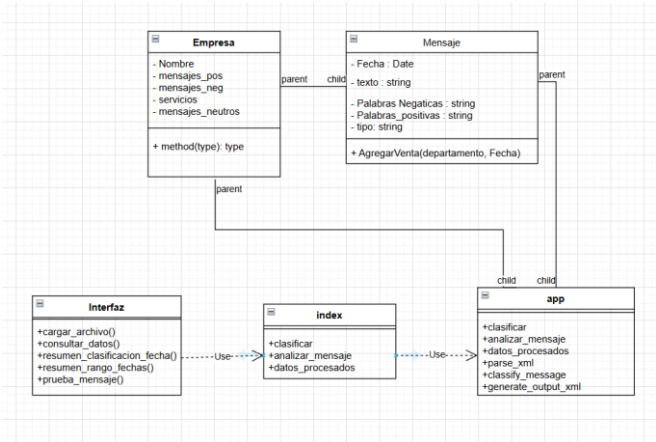
```
      <alias>préstamo</alias>
```

```
      <alias>crédito personal</alias>
```

```
    </servicio>
```

```
</servicios>
</empresa>
</empresas_analizar>
</diccionario>
<lista_mensajes>
  <mensaje>
    Lugar y fecha: Guatemala, 01/04/2022 15:01
    Usuario: user001@usac.edu
    Red social: Twitter
    El servicio de inscripción en la USAC fue excelente
    y muy satisfactorio.
  </mensaje>
  <mensaje>
    Lugar y fecha: Guatemala, 02/04/2022 12:15
    Usuario: user002
    Red social: Facebook
    Hoy abrí una cuenta en el Banco Nacional. El
    servicio fue bueno, aunque el proceso algo lento.
  </mensaje>
</lista_mensajes>
</solicitud_clasificacion>
```

Apéndice B: Diagramas de clases



Apéndice C: Diagrama de actividades

