
ANALIZADOR DE CONTENIDO EN REDES SOCIALES

202200349 – Irving Fernando Alvarado Asensio

Resumen

El proyecto se enfoca en la creación de una aplicación web que aprovecha Flask en el backend y Django en el frontend para realizar un análisis exhaustivo de los sentimientos y las estadísticas asociadas a los mensajes de redes sociales. Los usuarios tienen la tarea de cargar archivos XML que contienen tanto diccionarios de palabras positivas como negativas, junto con archivos XML que contienen los mensajes que desean someter al análisis. La aplicación se adentra en cada uno de estos mensajes, escudriñando palabra por palabra con el objetivo de discernir si el mensaje se percibe como positivo, negativo o neutral, todo basado en la coincidencia con los diccionarios previamente cargados. Este proyecto proporciona una solución sólida para el análisis de sentimientos y estadísticas en redes sociales. Permite a los usuarios comprender y evaluar mejor los mensajes publicados en diversas redes sociales, lo que facilita la toma de decisiones y la interpretación de las publicaciones hechas por algún grupo de personas.

Palabras clave

Flask, Django, Python, Web, Http

Abstract

The project focuses on the development of a web application that leverages Flask in the backend and Django in the frontend to perform a comprehensive analysis of sentiments and associated statistics related to social media messages. Users are tasked with uploading XML files that contain both positive and negative word dictionaries, as well as XML files containing the messages they wish to subject to analysis. The application delves into each of these messages, scrutinizing word by word with the aim of discerning whether the message is perceived as positive, negative, or neutral, all based on the match with the previously loaded dictionaries. This project provides a robust solution for sentiment analysis and social media statistics. It empowers users to better understand and evaluate the messages posted on various social media platforms, facilitating decision-making and the interpretation of posts made by a group of individuals

Keywords

Flask, Django, Python, Web, Http

Introducción

Este ensayo se centra en presentar una aplicación web que combina Flask en el backend y Django en el frontend, diseñada para llevar a cabo un análisis de sentimientos y estadísticas de mensajes en redes sociales. El objetivo principal de este ensayo es proporcionar una descripción detallada y un manual de usuario para comprender cómo funciona esta aplicación, desde la carga de archivos XML con diccionarios de palabras hasta el análisis de mensajes y la generación de estadísticas. También se proporcionará una visión general de las características de la aplicación, incluyendo la generación de gráficos que visualizan las tendencias de hashtags y menciones en un rango de fechas, así como la cantidad de mensajes positivos, negativos y neutros. En última instancia, este ensayo tiene como objetivo servir como un recurso práctico y útil para comprender el funcionamiento de la aplicación, lo que permitirá a los usuarios aprovechar al máximo esta herramienta.

Desarrollo del tema

a. Descripción de la aplicación

La aplicación desarrollada es una plataforma web que combina Flask en el backend y Django en el frontend, diseñada para realizar análisis de sentimientos y estadísticas en mensajes de redes sociales. Su objetivo principal es proporcionar a los usuarios una herramienta efectiva para evaluar y comprender mejor los mensajes publicados en diversas plataformas de redes sociales. Además, la aplicación ofrece la generación de estadísticas y gráficos que visualizan tendencias de hashtags y menciones en un

rango de fechas, así como la cantidad de mensajes con sentimientos positivos, negativos y neutros. Esto proporciona una visión completa de la percepción pública en las redes sociales y facilita la interpretación de los datos.

b. Carga de archivos

La carga de archivos XML en la aplicación se lleva a cabo a través de una interfaz fluida y sencilla en el frontend. Los usuarios pueden cargar archivos tanto de diccionarios de palabras positivas y negativas como de mensajes para su posterior análisis. Para cargar los archivos, se proporciona un formulario en el frontend que permite a los usuarios seleccionar los archivos deseados desde su computadora y, al hacer clic en los botones "Cargar Diccionario" o "Cargar Mensajes", se inicia el proceso de envío de datos al backend. El backend, basado en Flask, recibe estas solicitudes en los endpoints específicos: <http://127.0.0.1:5000/cargar-mensajes>, para los mensajes y <http://127.0.0.1:5000/cargar-diccionario> para los diccionarios y procede a procesar los archivos XML. En el caso de los diccionarios, actualiza los diccionarios internos de palabras, y en el caso de los mensajes, realiza el análisis de sentimientos y genera estadísticas. Como resultado, el backend genera un archivo de resumen que detalla las modificaciones realizadas y devuelve el nombre de este archivo como respuesta al frontend. Esta estructura colaborativa entre el frontend y el backend garantiza una experiencia de usuario intuitiva

y efectiva, al tiempo que permite un procesamiento de datos preciso y la generación de estadísticas útiles.



Figura 1. Interfaz para la carga de archivos.

Fuente: elaboración propia.

c. Análisis de mensajes

Cuando el backend recibe un archivo de mensajes, inicia un proceso detallado de análisis. En primer lugar, se realiza la normalización del texto, lo que implica la eliminación de tildes y la conversión a minúsculas para garantizar una consistencia en el procesamiento. Luego, se emplean expresiones regulares para extraer los hashtags y arrobas presentes en los mensajes, y se los separa del texto principal. Este paso es crucial para preparar los datos para un análisis estadístico posterior.

El backend también categoriza tanto los hashtags como los arrobas identificados en los mensajes. Esta categorización permitirá un análisis más profundo cuando se requieran estadísticas sobre tendencias de menciones y hashtags.

Posteriormente, el backend se concentra en el análisis de las palabras restantes en los mensajes que no son ni arrobas ni hashtags. Cada palabra se compara individualmente con los diccionarios de palabras positivas y negativas. En caso de que una palabra se

encuentre en el diccionario de palabras positivas, se suma un valor de 1; si se encuentra en el diccionario de palabras negativas, se suma un valor de -1. Este proceso continúa hasta que se haya evaluado cada palabra en el mensaje.

Finalmente, el backend realiza una evaluación del balance entre palabras positivas y negativas en el mensaje. Si la cantidad de palabras positivas es mayor que la de palabras negativas, el mensaje se considera positivo. Por el contrario, si la cantidad de palabras negativas supera a las positivas, se clasifica como negativo. En el caso de que ambas cantidades sean iguales, el mensaje se etiqueta como neutral. Este proceso se aplica a cada mensaje cargado, lo que permite una rápida y precisa categorización del sentimiento asociado a cada uno.

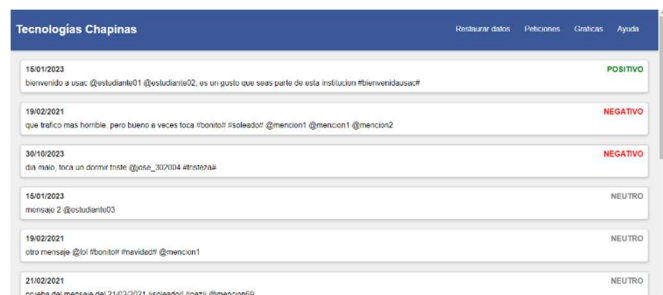


Figura 2. Mensajes cargados.

Fuente: elaboración propia.

d. Análisis de hashtags

Cuando los usuarios desean obtener estadísticas sobre el uso de hashtags en un rango de fechas, la aplicación permite realizar esta consulta de manera eficiente. El proceso se inicia desde el frontend, donde los usuarios ingresan una fecha de inicio y una fecha de fin para delimitar el período que desean analizar. La solicitud se envía al backend a través del endpoint `http://127.0.0.1:5000/consultar-hashtags`.

En el backend, se realiza una búsqueda en la base de datos de mensajes, filtrando todos los mensajes dentro del rango de fechas especificado por el usuario. Para cada mensaje en el rango de fechas, se extraen los hashtags presentes y se lleva un registro del número de veces que se ha utilizado cada uno. Este proceso se repite para cada día dentro del rango de fechas, lo que permite obtener un análisis detallado y desglosado por día.

El resultado se presenta al usuario a través del frontend en forma de "cartitas". Cada "cartita" representa un día dentro del rango de fechas y muestra los hashtags utilizados y la cantidad de veces que se han mencionado en ese día en particular. Este enfoque proporciona una visión clara y detallada del uso de hashtags a lo largo del tiempo.

Además, se genera una gráfica de barras que resume el análisis. La gráfica muestra todos los hashtags utilizados durante el período y la cantidad total de menciones en todo el rango de fechas. Esto permite a los usuarios identificar rápidamente los hashtags más populares y su frecuencia de uso.

Una característica adicional es la capacidad de imprimir tanto el reporte completo de "cartitas" como la gráfica en un archivo PDF si el usuario lo desea. Esto proporciona una forma conveniente de guardar y compartir los resultados del análisis.

En conjunto, esta funcionalidad de análisis estadístico de hashtags brinda a los usuarios una visión detallada y resumida del uso de hashtags en un rango de fechas especificado, lo que puede ser valioso para comprender las tendencias y dinámicas en las redes sociales.



Figura 3. Consulta de hashtags.

Fuente: elaboración propia.

e. Análisis de menciones

El análisis de menciones en la aplicación se realiza de manera análoga al análisis de hashtags, pero en lugar de rastrear el uso de hashtags, se centra en el seguimiento de menciones de usuarios en los mensajes. Los usuarios pueden solicitar este análisis desde el frontend proporcionando una fecha de inicio y una fecha de fin para definir el rango de fechas de interés.

En el backend, la solicitud del usuario se procesa en el endpoint <http://127.0.0.1:5000/consultar-usuarios>. El backend realiza una búsqueda en la base de datos de mensajes para encontrar todos los mensajes dentro del rango de fechas especificado. Para cada mensaje en ese período, se extraen y registran las menciones de usuarios.

El resultado se presenta al usuario en el frontend en forma de "cartitas" similares a las utilizadas en el análisis de hashtags. Cada "cartita" representa un día en el rango de fechas y muestra las menciones de usuarios realizadas en ese día y la cantidad de veces que se han mencionado.

Además, se genera una gráfica de barras resumiendo el análisis. La gráfica muestra

todos los usuarios mencionados durante el período y la cantidad total de menciones en todo el rango de fechas. Esto permite a los usuarios identificar rápidamente quiénes son los usuarios más mencionados y la frecuencia de sus menciones.

Como en el análisis de hashtags, los usuarios también tienen la opción de imprimir tanto el conjunto completo de "cartitas" como la gráfica en un archivo PDF, lo que facilita la conservación y el intercambio de los resultados del análisis.

En conjunto, esta funcionalidad de análisis estadístico de menciones brinda a los usuarios una visión detallada y resumida del uso de menciones de usuarios en un rango de fechas definido, lo que puede ser valioso para comprender quiénes son los usuarios más mencionados y cómo se relacionan en las redes sociales.



Figura 4. Consulta de menciones.

Fuente: elaboración propia.

f. Análisis de sentimientos

El análisis de sentimientos en la aplicación permite a los usuarios obtener una visión general de la cantidad de mensajes clasificados en tres categorías: positivos, negativos y neutros. Al igual que en las consultas anteriores, los usuarios pueden definir un rango de fechas proporcionando una fecha de inicio y una fecha de fin desde el frontend.

En el backend, la solicitud del usuario se procesa en el endpoint `http://127.0.0.1:5000/consultar-sentimientos`. El backend realiza una búsqueda en la base de datos de mensajes para encontrar todos los mensajes dentro del rango de fechas especificado.

El resultado se presenta al usuario en el frontend, organizado por fechas. La aplicación muestra la cantidad de mensajes positivos, negativos y neutros para cada día en el rango de fechas. Esto proporciona a los usuarios una visión clara de cómo se ha percibido el sentimiento en las redes sociales a lo largo del tiempo.

Además, los usuarios pueden acceder a un resumen gráfico que presenta de manera concisa el análisis de sentimientos. El gráfico resume la cantidad total de mensajes positivos, negativos y neutros en el rango de fechas especificado.

De manera similar a las consultas anteriores, los usuarios tienen la opción de imprimir tanto el conjunto completo de resultados como el gráfico en un archivo PDF, lo que facilita la conservación y el intercambio de los resultados del análisis.

En conjunto, esta funcionalidad de análisis de sentimientos brinda a los usuarios una visión detallada de cómo evolucionan las percepciones de sentimiento en las redes sociales dentro del rango de fechas definido. Esto puede ser valioso para comprender las tendencias de sentimiento y la dinámica de las conversaciones en línea.



Figura 5. Consulta de sentimientos.

Fuente: elaboración propia.

g. Estructura de la aplicación

- Frontend (Django):

Django se encarga de gestionar la capa de presentación y la interfaz de usuario. Proporciona una estructura de carpetas que contiene plantillas HTML, vistas y archivos estáticos, como hojas de estilo CSS y archivos JavaScript. El frontend interactúa directamente con los usuarios y recopila la entrada de datos, como la selección de archivos XML y las fechas para las consultas.

- Backend (Flask):

Flask, se utiliza en el backend de la aplicación. Gestiona la lógica de negocios, el procesamiento de solicitudes y respuestas, y la comunicación con la base de datos. Flask se encarga de la carga y procesamiento de archivos XML, el análisis de sentimientos y el análisis estadístico.

- Base de Datos:

La aplicación almacena datos esenciales en una base de datos. Esto puede incluir registros de mensajes, diccionarios de palabras, resultados de análisis y otros datos relevantes para el funcionamiento de la aplicación. Se utilizan archivos XML para almacenar datos.

- Generación de Gráficos:

Para la creación de gráficos que resumen los resultados del análisis estadístico, la aplicación utiliza bibliotecas, como ChartJs. Estos componentes generan gráficos visuales que se presentan al usuario en el frontend.

El frontend de Django proporciona una interfaz de usuario atractiva y fácil de usar. Los usuarios interactúan con la aplicación a través de formularios, botones y menús desplegables que les permiten cargar archivos XML, definir fechas de consulta y visualizar los resultados del análisis

En resumen, la aplicación se organiza en una estructura de frontend y backend, donde Django gestiona la presentación y la interacción con el usuario, mientras que Flask se encarga de la lógica de negocio y el procesamiento de datos. La base de datos almacena información importante, y los módulos de análisis.

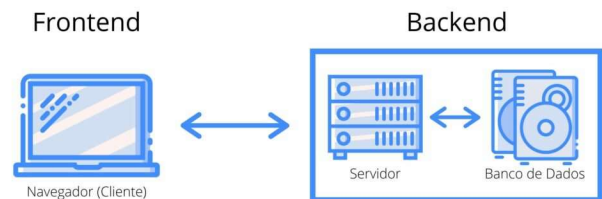


Figura 6. Flujo de la aplicación.

Fuente: (Fernandes, 2020).

Conclusiones

Esta aplicación web que emplea Flask y Django demuestra ser una herramienta integral y eficaz para el análisis de sentimientos y estadísticas en mensajes de redes sociales. Al permitir a los usuarios cargar archivos XML de diccionarios y mensajes, así como realizar consultas de hashtags, menciones y análisis de sentimientos en un rango de fechas, ofrece una solución valiosa para comprender las conversaciones en línea. La capacidad de generar gráficos y archivos PDF mejora la presentación de resultados. Este proyecto plantea cuestiones importantes, como cómo aprovechar de manera efectiva esta información en la toma de decisiones en redes sociales y cómo mejorar la precisión del análisis de sentimientos. La aplicación representa un avance significativo en el análisis de datos en línea y sugiere un potencial considerable para futuras investigaciones y desarrollos en este campo.

Referencias bibliográficas

Fernandes, H. M. (16 de Agosto de 2020). *¿Qué es un desarrollador frontend y que haces?* marques fernandes:
<https://marquesfernandes.com/es/tecnologia-es/que-es-un-frontend-desarrollador-y-que-hace/>