Prácticas de Datos abiertos y Visualización

Análisis de tweets sobre guerra Israel-Gaza

https://tweets-guerra.streamlit.app/

Laura Roca

Afi escuela de negocios

Máster en Data Science, Big Data e Inteligencia Artificial en Finanzas (MDSIAF)

2024

Índice de Contenido

Índice de Figuras

Extracción de datos

Importación de librerías

Se importaron las siguientes librerías:

asyncio: Permite la programación asíncrona.

twscrape: Biblioteca utilizada para acceder a la API de Twitter de forma asíncrona

pandas: Biblioteca para manipulación y análisis de datos en formato tabular.

Función principal para web scraping

Se utiliza la API TWSCRAPE: `https://github.com/vladkens/twscrape`.

Se define la función principal main() que inicializa la API de Twitter, realiza la autenticación, y realiza una búsqueda inicial (parámetro de búsqueda “guerra israel gaza”) con límite de 500 tweets considerados ’top’, de los cuales extrae el id del tweet, la ubicación donde se escribió el tweet, el nombre de pila del usuario del tweet, la fecha de publicación del tweet, el username del tweet y el contenido del tweet.

La funciím itera asincrónicamente sobre los resultados de la búsqueda, extrayendo datos y almacenándolos en las listas hasta llegar a los 2000 tweets.

```python
# Función que utiliza librería geopy para obtener el pais de la ubicación proporcionada.
# fuente: https://pypi.org/project/geopy/
def obtener_pais_ubi(ubicacion):
    geolocalizador = Nominatim(user_agent="localizacion")
    try:
        location = geolocalizador.geocode(ubicacion)
        if location:
            pais = location.address.split(",")[-1].strip()
        else:
            pais = 'sin ubicación'
        return {ubicacion: pais}
    except Exception as e:
        print(f"Error al obtener país para '{ubicacion}': {e}")
        return {ubicacion: 'sin ubicación'}


diccionario_paises = {}
tweets_ubicaciones = tweets_guerra["ubicacion"].unique()

for ubicacion in tweets_ubicaciones:
    diccionario_paises.update(obtener_pais_ubi(ubicacion))
```

Figura 1.1: Función para web scrapping

Tratamiento y preparación de los datos

Importación de librerías

Se importaron las siguientes librerías:

pandas: Manipulación y análisis de datos en formato tabular.

geopy: Proporciona herramientas para realizar geocodificación y otras operaciones relacionadas con la geolocalización.

googletrans: Traducción automática utilizando la API de Google Translate.

pycountry: Proporciona información sobre países, incluyendo códigos ISO y nombres.

requests: Realiza solicitudes HTTP.

pytz: Manejo de zonas horarias.

datetime: Manipulación de objetos de fecha y hora.

re: Búsqueda y manipulación de patrones en cadenas de texto.

nltk: Biblioteca de procesamiento de lenguaje natural.

Exploración de Datos Iniciales

Se muestra una vista previa de los primeros registros y se realizan comprobaciones iniciales sobre la existencia de datos nulos. Posteriormente, se llenan los valores nulos en la columna 'ubicacion' con la etiqueta 'sin ubicación'.

Cambio de ubicación por país

Se define una función que utiliza la biblioteca geopy para obtener el país a partir de la ubicación proporcionada en los tweets y se aplica esta función a cada ubicación única en los tweets. Los paises no encotrados en las ubicaciones, se les coloca 'sin ubicación.

```python
# Función que utiliza librería geopy para obtener el país de la ubicación proporcionada.
# fuente: https://pypi.org/project/geopy/
def obtener_pais_ubi(ubicacion):
    geolocalizador = Nominatim(user_agent="localizacion")
    try:
        location = geolocalizador.geocode(ubicacion)
        if location:
            pais = location.address.split(",")[-1].strip()
        else:
            pais = 'sin ubicación'
        return {ubicacion: pais}
    except Exception as e:
        print(f"Error al obtener país para '{ubicacion}': {e}")
        return {ubicacion: 'sin ubicación'}


diccionario_paises = {}
tweets_ubicaciones = tweets_guerra["ubicacion"].unique()

for ubicacion in tweets_ubicaciones:
    diccionario_paises.update(obtener_pais_ubi(ubicacion))
```

Figura 2.1: Función para encontrar el país de la ubicación.

Luego de ejecutar la función, se arregla manualmente algunas ubicaciones no encontradas por la función, con la finalidad de tener un set de datos más limpio. Este paso se debe omitir o ajustar para reproducir el código con diferentes bases de datos.

Traducción de país y tweet a español

Se define una función que utiliza la biblioteca googletrans para detectar el idioma de la ubicación y tweets y traducirlos a español, con la finalidad de tener toda la información en un mismo idioma. Las ubicaciones y tweets que la función no logra traducir, las deja igual.

```python
def traducir_a_espanol(texto):
    try:
        translator = Translator()
        traduccion = translator.translate(texto, src='auto', dest='es')
        return traduccion.text
    except Exception as e:
        print(f"Error: {e} al traducir el tweet: {texto}")
        return texto
```

Figura 2.2: Función para traducir a español

Antes de traducir los tweets a español, se define una función que utiliza expresiones regulares para limpiar los tweets, eliminando menciones, hashtags y emojis.

Luego de ejecutar las funciones, se arregla manualmente algunas ubicaciones mal traducidas, con la finalidad de tener un set de datos más limpio. Este paso se debe omitir o ajustar para reproducir el código con diferentes bases de datos.

Ajuste de hora según el país de ubicación

Para tener un análisis mas veraz y contextual de los tweets, se realiza el cambio de hora según la zona horaria que pertenece cada país de la ubicación. Para poder realizar esto, primero se debe encontrar el código ISO-2 de cada país. Por lo tanto, se define una función que utiliza la biblioteca pycountry para obtener los códigos ISO-2 de los países y se aplica a la lista de países presentes en los tweets. Los países que no dispongan de un código asignado se traducirán al inglés y se realizará un intento para encontrar el código correspondiente. En caso de no obtener éxito, se registrará la frase 'No se encontró código'.

```python
def obtener_codigos_paises(nombres_paises):
    codigos_paises = {}

    for nombre_pais in nombres_paises:
        try:
            resultado_busqueda = pycountry.countries.search_fuzzy(nombre_pais)
            pais = resultado_busqueda[0].alpha_2
            codigos_paises[nombre_pais] = pais
        except LookupError:
            try:
                translator = Translator()
                pais_en = translator.translate(nombre_pais, src='es', dest='en')
                pais = pycountry.countries.search_fuzzy(pais_en.text)[0]
                codigos_paises[nombre_pais] = pais.alpha_2
            except LookupError:
                codigos_paises[nombre_pais] = "no se encontró código"

    return codigos_paises
```

Figura 2.3: Función para encontrar código ISO-2 del país

Una vez, se tienen los códigos ISO-2 de cada país, se define una función que utiliza la biblioteca pytz para ajustar la hora de los tweets según la zona horaria de los codigos ISO-2 de cada país. Las ubicaciones que no se les encuentre zona horaria, se les coloca la zona horaria de España.

```python
def ajustar_hora(row):
    pais = row['codigo_pais']
    hora = row['fecha']
    try:
        zona_horaria = pytz.country_timezones[pais][0]
    except KeyError:
        zona_horaria = pytz.country_timezones["ES"][0]

    dt = datetime.strptime(hora, '%Y-%m-%d %H:%M:%S%z')
    dt = dt.astimezone(pytz.timezone(zona_horaria))

    return dt.strftime('%Y-%m-%d %H:%M:%S')
```

Figura 2.4: Función para ajustar hora según zona horaria

Transformaciones Temporales

Se realizan diversas transformaciones en las columnas de fechas y horas para obtener información adicional, como mes, año, día de la semana y período del día.

Obtención de Coordenadas Geográficas de Países

Se define una función que utiliza la biblioteca requests para realizar web scraping a la pagina 'https://restcountries.com/v2/alpha/codigo_iso2', donde codigo_iso2 corresponde al código del país de cada ubicación y así, obtener las coordenadas geográficas de los países.

```
def obtener_coordenadas_iso2(codigos_iso2):
        resultados = []

        for codigo_iso2 in codigos_iso2:
            try:
                    url = f"https://restcountries.com/v2/alpha/{codigo_iso2}"
                    respuesta = requests.get(url)

                    if respuesta.status_code == 200:
                        datos_pais = respuesta.json()
                        longitud = datos_pais['latlng'][1]
                        latitud = datos_pais['latlng'][0]

                        resultados.append({
                            'codigo_pais': codigo_iso2,
                            'Latitud': latitud,
                            'Longitud': longitud
                        })
            except Exception as e:
                    print(e)
        df = pd.DataFrame(resultados)
        return df
```

Figura 2.5: Función para obtener coordenadas de países

Análisis de sentimientos

Se define una función que utiliza la biblioteca nltk para realizar análisis de sentimientos en los tweets y clasificarlos en positivo, negativo o neutro.

La función invoca el método polarity_scores del objeto analizador para obtener un diccionario de puntuaciones de sentimiento para el texto, que incluye valores para las dimensiones de sentimiento positivo, negativo y neutral.

Posteriormente, la función examinará si la puntuación compuesta es mayor que 0.05, en tal caso, se considerará que el sentimiento es positivo. Si la puntuación es menor que -0.05, se clasificará como negativo. En caso contrario, se determinará que el sentimiento es neutro.

```
def analisis_sentimientos(texto):
    analizador = SentimentIntensityAnalyzer()

    sentiment_score = analizador.polarity_scores(texto)

    if sentiment_score['compound'] >= 0.05:
        return "Positivo"
    elif sentiment_score['compound'] <= -0.05:
        return "Negativo"
    else:
        return "Neutral"
```

Figura 2.6: Función para realizar análisis de sentimientos

Justificación de las visualizaciones

Gráficas de análisis territorial

Gráfico de burbujas en mapa

Este gráfico proporciona una vista geográfica de la distribución de los tweets relacionados con el conflicto Israel-Gaza, resaltando la cantidad de tweets por ubicación y su proporción. Muestra la concentración de tweets en diferentes regiones y países, proporcionando una perspectiva geoespacial de la discusión en Twitter, además, facilita la identificación de áreas con mayor actividad y resalta la diversidad geográfica de la conversación.

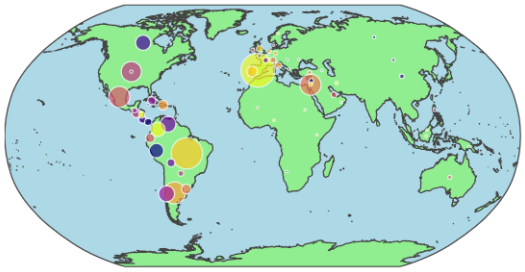

Figura 3.1: Gráfico de burbujas en mapa

Mapa de calor

Este mapa de calor destaca visualmente la intensidad de la discusión sobre el conflicto en diferentes ubicaciones mediante el color, con áreas más oscuras representando una mayor concentración de tweets. Permite una fácil identificación de las regiones más activas en términos de discusión, resaltando las áreas de mayor interés o impacto.

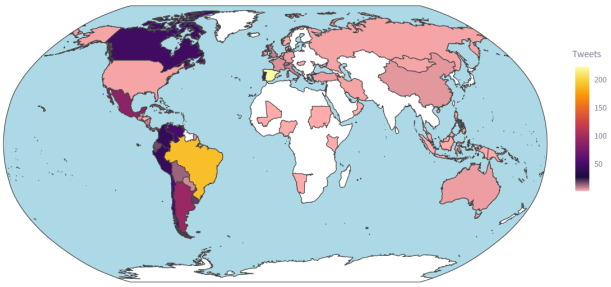

Figura 3.2: Mapa de calor

Gráfico de barras - países

Este gráfico de barras muestra la cantidad de tweets por ubicación en un formato ordenado, proporcionando una visión rápida de las ubicaciones más activas en la discusión. Facilita la comparación entre diferentes ubicaciones, permitiendo identificar rápidamente las áreas más relevantes en términos de discusión sobre el conflicto.

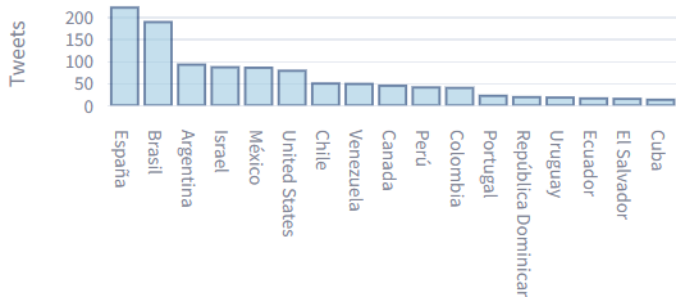

Figura 3.3: Gráfico de barras- países

Gráficas de análisis temporal

Gráfico de Línea Temporal - Fechas

Este gráfico muestra la evolución temporal de la cantidad de tweets relacionados con el conflicto. Permite identificar patrones de actividad a lo largo del tiempo, destacando picos o tendencias en la discusión.

Gráfico de Barras - Días de la Semana

Muestra la distribución de tweets a lo largo de los días de la semana. Facilita la identificación de los días de la semana con mayor o menor actividad en la discusión.

Gráfico de Línea Temporal - Horas

Ofrece una visión de la variación horaria en la actividad de tweets. Permite identificar las horas del día con mayor participación en la discusión.

Gráfico de Barras - Períodos del día

Muestra la distribución de tweets en diferentes períodos del día (mañana, tarde, noche). Facilita la comprensión de cuándo ocurren la mayoría de las interacciones en relación con el conflicto.

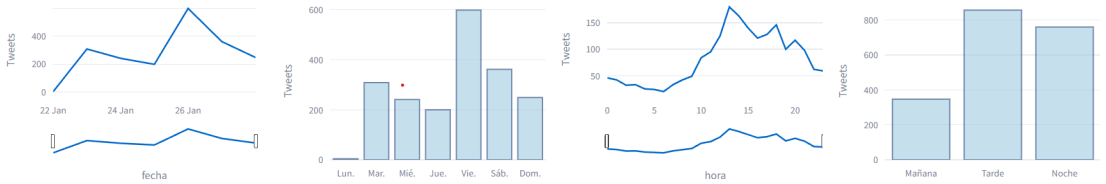

Figura 3.4: Gráficas temporales

Mapa de calor dinámico

El gráfico de densidad dinámico tiene como objetivo mostrar la distribución de tweets sobre el conflicto entre Israel y Gaza en diferentes países a lo largo del tiempo. Permite observar la variación de la actividad de tweets en distintos países a medida que transcurre el tiempo, proporcionando una perspectiva dinámica y geoespacial de la discusión.

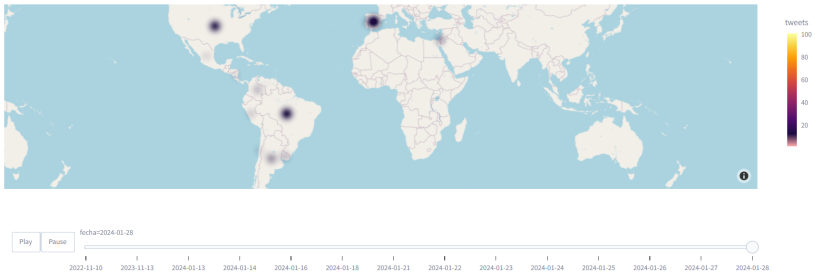

Figura 3.5: Mapa de calor dinámico

Gráficas de análisis de sentimientos

Gráfico de Barras - Sentimientos

Muestra la distribución de los diferentes sentimientos (positivo, neutral, negativo) en los tweets. Facilita la comparación entre las categorías de sentimientos y destaca la prevalencia de un sentimiento particular en la discusión.

Gráfico Donut - Sentimientos

Representa la distribución proporcional de los sentimientos en un formato circular, resaltando la proporción relativa de cada sentimiento. La representación en forma de donut permite una fácil identificación de la proporción de cada sentimiento en relación con el total.

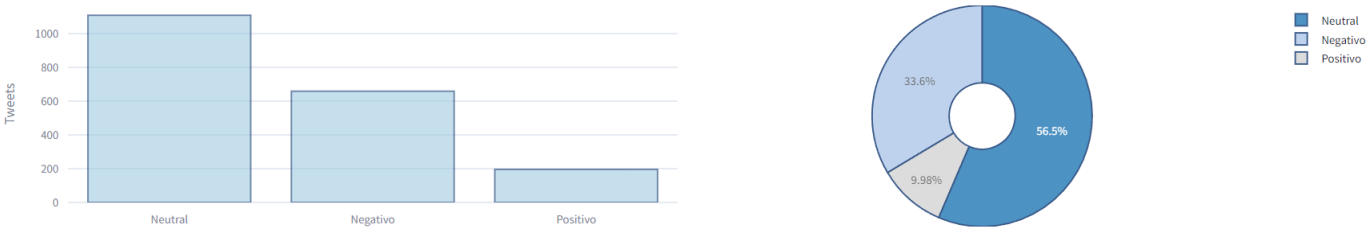

Figura 4.1: Gráficos de sentimientos

Nube de palabras

Representa visualmente las palabras más frecuentes en los tweets, donde el tamaño de la palabra indica su frecuencia de aparición. Facilita la identificación rápida de las palabras clave y temas dominantes en la discusión.

Figura 4.2: Nube de palabras

Conclusiones

Análisis territorial

Se destaca la presencia activa de diversos países, como España, Brasil, Argentina, Israel, México y Estados Unidos, indicando un interés global en el conflicto. Además, la inclusión de naciones del continente americano sugiere un interés particular en la región, posiblemente relacionado con sensibilidades políticas o expresiones de solidaridad.

El involucramiento local en Israel es evidente, subrayando la importancia del conflicto en la conversación interna del país.

Análisis temporal

Se destaca que el día con mayor cantidad de tweets populares relacionados con el conflicto Israel-Gaza fue el 26 de enero de 2024. Además, se observa una tendencia significativa los viernes, siendo este el día de la semana con la mayor frecuencia de tweets. En cuanto al horario, se registra un pico a las 13:00, destacando la tarde como el periodo del día con la mayor concentración de tweets populares sobre el mencionado conflicto. Además, se observa que con el paso del tiempo, se diversifica la participación de países en la discusión sobre el conflicto.

Análisis de sentimientos

La mayoría de los tweets, un 56.5 %, son clasificados como neutros, lo que sugiere que gran parte de la conversación tiende a expresar información de manera objetiva. Por otro lado, un 33.6 % de los tweets son negativos, indicando que hay una proporción considerable de contenido que refleja opiniones desfavorables. En contraste, un 9.98 % de los tweets son positivos, lo que sugiere que, a pesar de la naturaleza conflictiva del tema, existe un segmento de la conversación que manifiesta opiniones favorables o esperanzadoras.

La nube de palabras resalta términos, como ’guerra’, ’Gaza’, ’Israel’, ’Hamas’, ’genocidio’, ’justicia’ y ’ataques’, estos reflejan un enfoque intenso en aspectos conflictivos y éticos. La inclusión de términos como ’alto’ podría indicar un llamado a cese de fuego.