

## Proyecto PROMPT

10/11/2023

Presentado por: Juan Pablo Durán Martín

### Modelo de análisis de sentimientos de posts en Reddit para obtener nube de palabras de posts negativos

#### Resumen

En este informe se resume el desarrollo llevado a cabo con ingeniería de prompt y chatGPT para configurar la API de Reddit y obtener los posts que contengan un string en específico, en este caso será *"Bancolombia"*, y posteriormente por un modelo analizar el sentimiento del post (ya sea positivo, negativo o neutro) y con base a los posts negativos obtener una nube de palabras para implementar alternativas sobre cuáles son las problemáticas actuales o los disgustos de las personas de manera actualizada.

#### Implementación de API de Reddit

Pregunta:




Hola ¿Sabes como puedo implementar pysentimiento en google colab y analizar con la API de Reddit las publicaciones que contengan la palabra Bancolombia?

Respuesta:

1. Instala PRAW en Google Colab ejecutando el siguiente comando:


```
python
```

 Copy code

```
!pip install praw
```

1. Crea una aplicación en Reddit para obtener las credenciales necesarias. Sigue los pasos mencionados anteriormente en el paso 1 de la respuesta anterior.
2. Importa las bibliotecas necesarias y configura la autenticación con las credenciales de Reddit:

python

 Copy code

```
import praw


# Configura las credenciales de Reddit
reddit_credentials = {
    "client_id": "TU_CLIENT_ID",
    "client_secret": "TU_CLIENT_SECRETO",
    "user_agent": "Nombre de tu aplicación",
    "username": "Tu nombre de usuario en Reddit",
    "password": "Tu contraseña en Reddit"
}

# Inicializa el cliente de Reddit
reddit = praw.Reddit(
    client_id=reddit_credentials["client_id"],
    client_secret=reddit_credentials["client_secret"],
    user_agent=reddit_credentials["user_agent"],
    username=reddit_credentials["username"],
    password=reddit_credentials["password"]
)
```

Asegúrate de reemplazar las cadenas `"TU_CLIENT_ID"`, `"TU_CLIENT_SECRETO"`, `"Nombre de tu aplicación"`, `"Tu nombre de usuario en Reddit"`, y `"Tu contraseña en Reddit"` con las credenciales y datos de tu aplicación de Reddit.

1. Utiliza la biblioteca PRAW para buscar publicaciones que contengan la palabra "Bancolombia". Aquí tienes un ejemplo:

python

 Copy code

```
# Realiza una búsqueda en Reddit
subreddit = reddit.subreddit("all") # Puedes especificar un subreddit espec
posts = subreddit.search("Bancolombia", limit=10)

# Imprime las publicaciones encontradas
for post in posts:
    print(f"Título: {post.title}")
    print(f"Contenido: {post.selftext}")
    print(f"Puntuación: {post.score}")
    print(f"Fecha de creación: {post.created_utc}\n")
```

## Implementacion de modelo de analisis de sentimiento

Pregunta:



¿Ahora como realizo el análisis de sentimiento?

Respuesta:



Para realizar un análisis de sentimiento en el contenido de las publicaciones de Reddit, puedes utilizar `pysentimiento`. Aquí hay un ejemplo de cómo hacerlo:

1. Asegúrate de que has instalado `pysentimiento` en tu entorno de Google Colab como se mencionó anteriormente:

python

Copy code

```
!pip install pysentimiento
```

1. Importa la biblioteca `pysentimiento` en tu notebook de Colab:

python

Copy code

```
import pysentimiento
```

1. Utiliza la función `pysentimiento.sentiment` para analizar el sentimiento del contenido de las publicaciones que obtuviste de Reddit. Puedes aplicar esta función a cada publicación en tu lista de publicaciones. Aquí hay un ejemplo:

python

Copy code

```
# Función para realizar el análisis de sentimiento en el contenido de una publicación
def analyze_sentiment(text):
    sentiment_analyzer = pysentimiento.SentimentAnalyzer(lang="es")
    result = sentiment_analyzer.predict(text)
    return result

# Analiza el sentimiento de las publicaciones
for post in posts:
    title_sentiment = analyze_sentiment(post.title)
    content_sentiment = analyze_sentiment(post.selftext)

    print(f"Titulo de la publicación: {title_sentiment['text']}")
    print(f"Sentimiento del titulo: {title_sentiment['sentiment']}\n")

    print(f"Contenido de la publicación: {content_sentiment['text']}")
    print(f"Sentimiento del contenido: {content_sentiment['sentiment']}\n")
```

Volver a generar

En este ejemplo, hemos definido una función `analyze_sentiment` que utiliza `pySentimiento` para realizar el análisis de sentimiento en un texto dado. Luego, aplicamos esta función a los títulos y el contenido de las publicaciones que obtuviste de Reddit y mostramos los resultados.

Recuerda que la precisión del análisis de sentimiento depende de la calidad del modelo de lenguaje subyacente y de la calidad del texto de las publicaciones en Reddit. Además, ajusta el código según tus necesidades específicas y considera manejar posibles excepciones y errores que puedan ocurrir durante la ejecución del código.

## Visualización de grafico de barras con clasificación de sentimientos en posts

Pregunta:



¿Cómo puedo visualizar los resultados en un grafico donde pueda ver la cantidad de posts si es positivo, negativo o neutro? Además de detectar cuales son los comentarios mas repetidos

Respuesta:



Para visualizar los resultados del análisis de sentimiento en un gráfico que muestre la cantidad de posts positivos, negativos y neutros, así como para detectar los comentarios más repetidos, puedes seguir estos pasos:

1. Crear listas para almacenar los resultados del análisis de sentimiento y los comentarios:

python

Copy code

```
positive_posts = []
negative_posts = []
neutral_posts = []

all_comments = []

# Analiza el sentimiento de las publicaciones y almacena los resultados en 1
for post in posts:
    title_sentiment = analyze_sentiment(post.title)
    content_sentiment = analyze_sentiment(post.selftext)


    if title_sentiment['sentiment'] == 'positivo' or content_sentiment['sentiment'] == 'positivo':
        positive_posts.append(post)
```

```
elif title_sentiment['sentiment'] == 'negativo' or content_sentiment['se  
negative_posts.append(post)  
else:  
    neutral_posts.append(post)  
  
# Almacena los comentarios en la lista de todos los comentarios  
all_comments.extend(post.comments.list())
```

1. Para visualizar la cantidad de publicaciones positivas, negativas y neutras en un gráfico de barras, puedes utilizar la biblioteca `matplotlib` para crear el gráfico. Asegúrate de tener instalada la biblioteca y luego utiliza el siguiente código:

1. Para visualizar la cantidad de publicaciones positivas, negativas y neutras en un gráfico de barras, puedes utilizar la biblioteca `matplotlib` para crear el gráfico. Asegúrate de tener instalada la biblioteca y luego utiliza el siguiente código:

python

 Copy code

```
import matplotlib.pyplot as plt  
  
# Crear un gráfico de barras  
sentiments = ['Positivo', 'Negativo', 'Neutro']  
counts = [len(positive_posts), len(negative_posts), len(neutral_posts)]  
  
plt.bar(sentiments, counts)  
plt.xlabel('Sentimiento')  
plt.ylabel('Número de Publicaciones')  
plt.title('Análisis de Sentimiento de las Publicaciones')  
plt.show()
```

Esto generará un gráfico de barras que muestra la cantidad de publicaciones con sentimientos positivos, negativos y neutros.

## Correccion de clasificacion de posts

Inicialmente se estaba realizando el análisis de sentimiento con base al título y el contenido, pero debido a que se estaba visualizando que muchos resultaban “Neutro” es necesario realizar un promedio con base al sentimiento de los comentarios del post.

Pregunta:



Mi idea con el modelo es revisar el titulo del post, el texto y los comentarios

for post in posts:

```
print(f"Post title: {post.title}\n")
```

```
print(f"Post body: {post.selftext}\n")
```

```
post.comments.replace_more(limit=None) # para obtener todos los comentarios
```

```
for comment in post.comments.list():
```

```
    analysis = analyzer.predict(comment.body) # Analizar el comentario
```

```
    print(f"Comment: {comment.body}")
```

```
    print(f"Sentiment: {analysis.output}\n")
```

Con esto la idea es clasificar con base a los comentarios si el post es positivo, neutro o negativo. Ver de manera grafica los resultados de la clasificación y visualizar cuales son los comentarios que mas se repiten




Para clasificar si el post es positivo o negativo debe ser con base al resultado de post.selftext, pero en caso de ser neutro, debe ser con base al mayor numero de comentarios positivos, negativos o neutros

Respuesta:



Entendido, si deseas clasificar un post como positivo o negativo en función del contenido del propio post, pero como neutro en función de los comentarios, puedes ajustar el código de la siguiente manera:

python

 Copy code

```
from pysentimiento import create_analyzer
import matplotlib.pyplot as plt
from collections import Counter

# Configurar el analizador de sentimientos
transformers.logging.set_verbosity(transformers.logging.ERROR)
analyzer = create_analyzer(task="sentiment", lang="es")

# Listas para almacenar los resultados
posts_classification = {"Positivo": [], "Negativo": [], "Neutro": []}
all_comments = []


# Realizar el análisis de sentimiento y recolectar los comentarios
for post in posts:
    title_sentiment = analyzer.predict(post.title).output
    content_sentiment = analyzer.predict(post.selftext).output
```

```
post_sentiments = {"Positivo": 0, "Negativo": 0, "Neutro": 0}

post.comments.replace_more(limit=None) # Para obtener todos los comentarios
for comment in post.comments.list():
    comment_analysis = analyzer.predict(comment.body)
    all_comments.append(comment.body)

    if comment_analysis.output == 'POS':
        post_sentiments["Positivo"] += 1
    elif comment_analysis.output == 'NEG':
        post_sentiments["Negativo"] += 1
    else:
        post_sentiments["Neutro"] += 1

# Clasificar el post en función del contenido y los comentarios
if content_sentiment == 'POS':
    posts_classification["Positivo"].append(post)
elif content_sentiment == 'NEG':
    posts_classification["Negativo"].append(post)
else:
    max_sentiment = max(post_sentiments, key=post_sentiments.get)
    posts_classification[max_sentiment].append(post)
```

 Volver a código

```
# Crear un gráfico de barras para visualizar la clasificación
sentiments = ["Positivo", "Negativo", "Neutro"]
counts = [len(posts_classification["Positivo"]), len(posts_classification["Negativo"]), len(posts_classification["Neutro"])]

plt.bar(sentiments, counts)
plt.xlabel("Sentimiento")
plt.ylabel("Número de Publicaciones")
plt.title("Clasificación de Sentimientos de los Posts")
plt.show()
```

## Desarrollo nube de palabras

Pregunta:



En lugar de contar la frecuencia de los comentarios, se podría con base a los comentarios negativos, analizar que palabras se repiten mas o procesarlos con una IA y entender que quieren decir

Respuesta:



Claro, en lugar de contar la frecuencia de los comentarios, puedes analizar los comentarios negativos para identificar las palabras que se repiten más y procesarlos con el objetivo de comprender su contenido. Aquí te muestro cómo hacerlo utilizando el análisis de palabras clave en los comentarios negativos:

```
# Procesar los comentarios negativos para identificar palabras clave
nltk.download('stopwords')
stop_words = set(stopwords.words("spanish"))
negative_comments_text = " ".join(negative_comments)
wordcloud = WordCloud(stopwords=stop_words, background_color='white', width=

# Visualizar la nube de palabras clave
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title("Nube de Palabras Clave en Comentarios Negativos")
plt.show()
```

## Filtrado de posts en español

Debido a que se obtenían varios posts en inglés y otros idiomas, es necesario filtrarlos y dejar únicamente los que estén en idioma español.

Pregunta:



¿Hay manera de filtrar los posts obtenidos c que sean solo en español




Respuesta:



Para filtrar los posts obtenidos de Reddit y asegurarte de que sean solo en español, puedes realizar una verificación del idioma del contenido de cada post utilizando bibliotecas de procesamiento de lenguaje natural como `langdetect` o `textblob`. A continuación, te mostraré un ejemplo utilizando `langdetect` para verificar si el contenido del post está en español:

Primero, asegúrate de instalar la biblioteca `langdetect` si aún no lo has hecho:


python

 Copy code

```
!pip install langdetect
```

Luego, puedes modificar tu código de la siguiente manera:

python

 Copy code

```
import praw
from langdetect import detect
```

```
# Filtra los posts que están en español
for post in posts:
    try:
        # Verifica si el contenido del post está en español
        if detect(post.title) == 'es' or detect(post.selftext) == 'es':
            print(f"Título: {post.title}")
            print(f"Contenido: {post.selftext}")
            print(f"Puntuación: {post.score}")
            print(f"Fecha de creación: {post.created_utc}\n")
    except:
        pass
```

En este código, se utiliza `langdetect` para verificar si el contenido del título y el cuerpo del post está en español (`'es'`). Si el contenido está en español, se imprime el post. Ten en cuenta que la detección de idioma no siempre es perfecta, y algunos posts pueden estar en español incorrectamente identificados. Puedes ajustar este enfoque según tus necesidades específicas.

---

## Conclusiones

Gracias a la ingeniería de prompt fue posible construir una solución a la problemática planteada al inicio, el resultado final se encuentra en el notebook: Proyecto\_final.ipynb