



Universidad de Costa Rica

Escuela de Matemáticas
Herramientas de Ciencias de Datos II
CA-0305

Trabajo Escrito - Proyecto

**Análisis de chats con Python:
Extracción de patrones y sentimientos en conversaciones.**

PROFESOR

Luis Alberto Juárez Potoy

ESTUDIANTES

Javier Hernández Navarro - C13674

Luis Amey Apuy - C20470

Anthony Jiménez Navarro - C24067

I - 2024

Índice

1	Introducción	1
2	Metodología	2
3	Resultados	4
4	Conclusiones	8
5	Recomendaciones y Extensión de Resultados	9
6	Anexos	9
	Referencias	I

1. Introducción

La idea principal del proyecto es desarrollar un sistema que permita transformar y analizar conversaciones de chat de WhatsApp y Telegram. Este proyecto busca aprovechar tanto las capacidades de varias librerías de R, para tratar los datos obtenidos de la forma adecuada, así como librerías de Python utilizadas para realizar la traducción de cadenas de texto del español al inglés, y al mismo tiempo llevar a cabo el complejo proceso del análisis de sentimientos.

El primer paso en este proceso es convertir los formatos de los chats a uno que sea más adecuado para el análisis. Los mensajes descargados de WhatsApp y Telegram vienen en formatos que no son directamente utilizables para el análisis de sentimientos deseado. Por ejemplo, los mensajes de WhatsApp suelen exportarse en archivos de texto plano con un formato específico, mientras que los chats de Telegram se exportan en formato JSON. Cada uno de estos formatos tiene sus particularidades que deben ser manejadas adecuadamente para extraer la información relevante y estructurar esta de una manera que facilite su análisis posterior.

Una vez que los datos se encuentran en el formato adecuado, el siguiente problema que se presenta es la traducción de los mensajes. Dado que las herramientas más avanzadas de procesamiento de lenguaje natural (NLP) están optimizadas para trabajar con textos en inglés, es necesario realizar la traducción de los mensajes de español a inglés. Para llevar a cabo esta tarea, se va a hacer uso de una librería de traducción automática que emplea el servicio de GoogleTranslator. Esta herramienta permitirá traducir de manera eficiente y precisa los grandes volúmenes de texto presentes en los chats.

Con los datos traducidos, existe la posibilidad de llevar a cabo el análisis de sentimientos, así como obtener una serie de características o patrones de los chats. Este análisis se llevará a cabo utilizando diversas librerías de Python dedicadas al procesamiento de lenguaje natural, tales como Natural Language Toolkit (NLTK) y TextBlob. NLTK proporciona una multitud de herramientas para el procesamiento de texto, incluyendo tokenización, etiquetado de partes del discurso, y, para el caso específico del proyecto, análisis de sentimientos mediante el uso de lexicones preentrenados como VADER, que es particularmente adecuado para analizar texto tanto en redes sociales como chats. TextBlob, por su parte, ofrece una interfaz de programación de aplicaciones (API) simple y accesible para realizar análisis de sentimientos.

El análisis de sentimientos resulta una herramienta considerablemente útil en el área del procesamiento del lenguaje natural, con aplicaciones que abarcan diversas áreas e industrias. Por ejemplo, puede utilizarse para evaluar la satisfacción de los cliente con un producto o servicio específico a través de su comportamiento en redes sociales con respecto al elemento de estudio, mejorando así las estrategias de atención al cliente y marketing. También puede aplicarse en el análisis de tendencias de mercado, permitiendo que las empresas tengan una mayor comprensión de la percepción que tienen los usuarios y públicos meta con respecto a sus productos o servicios. Además, en el ámbito de la salud mental, el análisis de sentimientos permita detectar cambios en el estado emocional de pacientes de forma prematura al realizar un análisis de la forma en que interactúan en diversas redes sociales, lo que se muestra como una herramienta complementaria en el monitoreo y apoyo psicológico.

El proyecto presente no solo busca transformar y analizar conversaciones de chat, sino también demostrar cómo las herramientas avanzadas de procesamiento de lenguaje natural pueden ser utilizadas de manera efectiva para obtener información de valor de datos textuales.

2. Metodología

El proyecto se desarrolló utilizando los lenguajes de programación R y Python, combinando así dos de los lenguajes cubiertos durante el curso. La integración de ambos lenguajes permitió aprovechar las fortalezas individuales de cada uno para realizar tareas específicas de limpieza, procesamiento y análisis de datos. A continuación, se describe detalladamente el proceso y las herramientas empleadas.

Inicialmente, se cargó la configuración requerida en R, lo cual incluye varias librerías esenciales. La librería *readr* se utilizó para leer datos en formato CSV o TSV de manera eficiente, lo cual es necesario para procesar las conversaciones descargadas de WhatsApp. Esta librería es especialmente útil debido a su capacidad para manejar grandes volúmenes de datos de forma adecuada y su flexibilidad en el manejo de diferentes delimitadores y distintos formatos de archivos.

Por otro lado, la librería *jsonlite* actúa como un analizador o "parser", así como también un generador de archivos .json para R, adecuado para manejar las conversaciones obtenidas de Telegram. Los archivos JSON son comúnmente utilizados para almacenar datos estructurados, y *jsonlite* facilita la conversión de estos archivos en data frames de R, permitiendo una manipulación y análisis más directo y sencillo.

El conjunto de librerías conocido como *tidyverse* también son de utilidad para el desarrollo del proyecto. *tidyverse* es una colección de paquetes de R diseñados para la ciencia de datos. Incluye herramientas para manipulación de datos (*dplyr*), visualización (*ggplot2*), importación de datos (*readr*), entre otras. Se utilizó *tidyverse* para dar un formato adecuado a los data frames de chat mediante la adecuada limpieza y transformación de datos.

La librería *reticulate* permite la interoperabilidad entre R y Python, y dada la naturaleza del proyecto, esta librería fue fundamental para combinar ambos lenguajes de programación. *reticulate* facilita la ejecución de código Python dentro de un script de R y viceversa, lo cual fue necesario para el desarrollo del proyecto utilizando ambos lenguajes, ya que Python resultaba más útil para el procesamiento de lenguaje natural (NLP).

Además, se definieron funciones en R para la limpieza de datos de los archivos de WhatsApp y Telegram, otorgándoles el formato necesario para realizar un análisis de sentimientos estandarizado para los dos data frames disponibles. Estas funciones incluyeron operaciones básicas de preprocesamiento de texto, como la eliminación de caracteres especiales, el manejo tanto de fechas como horas, y la normalización del texto.

Se establecieron dos funciones principales para la extracción de datos de los chats. La función *scrapW()* realiza el scraping de un archivo .txt de un chat de WhatsApp. Esta función se ajusta al formato específico de exportación, corrigiendo problemas de formato de tiempo (24 vs. 12 horas) y otros detalles específicos de la exportación del dispositivo específico utilizado. La función *scrapT()* realiza el scraping de un archivo .json de un chat de Telegram exportado desde la aplicación de escritorio de Telegram. Ambas funciones fueron diseñadas para extraer de manera eficiente y adecuada la información relevante de los archivos de chat.

Una vez establecida la configuración inicial, se aplicó la función *scrapW()* al archivo .txt, asignando el data frame resultante a la variable *gen*. De manera similar, se aplicó *scrapT()* al archivo .json, asignando el resultado a la variable *telegram*. Estos data frames contenían las conversaciones extraídas, listas para ser procesadas y analizadas.

Un desafío importante fue que las conversaciones estaban en español, mientras que las librerías de procesamiento de lenguaje natural (NLP) están desarrolladas mayormente para trabajar en inglés. Para resolver esto, se creó un módulo en Python con una clase llamada *AnalisisTexto*, la cual incluye varios métodos útiles. El método *init()* inicializa la clase, preparando el entorno para el procesamiento de datos. Los métodos *get()* y *set()* permiten obtener y establecer valores dentro de la clase, proporcionando flexibilidad en el manejo de datos.

El método `str()` representa el objeto como una cadena de texto, facilitando la visualización y depuración de datos. El método `traducir()` utiliza GoogleTranslator para traducir los mensajes de español a inglés, retornando un nuevo data frame traducido. El método `guardar()` permite guardar el data frame actual como un archivo .csv, asegurando que los datos traducidos se conserven de manera adecuada. Finalmente, el método `leer()` carga un archivo .csv y lo convierte a data frame, permitiendo la reutilización de datos previamente procesados.

Para el procesamiento y análisis de los datos en Python, se utilizaron varias librerías. Pandas se empleó para la manipulación y análisis de datos, facilitando la transformación de datos en estructuras tabulares como DataFrames. Esta librería es crucial para la organización y limpieza de los mensajes de chat, permitiendo operaciones como la filtración, transformación y agregación de datos.

nlk (Natural Language Toolkit) proporcionó herramientas para trabajar con texto, incluyendo tokenización y análisis de sentimientos con el léxico *VADER*. *VADER* es una herramienta de análisis de sentimientos que es particularmente adecuada para textos encontrados comúnmente en redes sociales como mensajes de chat. *deep_translator* permitió traducir texto del español al inglés de manera eficiente, utilizando el traductor de Google para convertir los mensajes. *TextBlob* facilitó tareas de procesamiento de lenguaje natural, este proporciona una API simple para llevar a cabo la tarea del análisis de sentimientos. Finalmente, Las librerías no fueron utilizadas en su totalidad, sino que mas bien se importaron componentes específicos como *SentimentIntensityAnalyzer* de *nlk.sentiment.vader* para analizar la intensidad de los sentimientos en los textos, GoogleTranslator de *deep_translator* para la traducción de mensajes, *TextBlob* si fue importado en su totalidad, esto para hacer uso de la mayoría de capacidades para facilitar el procesamiento de texto y análisis de sentimientos.

Se estableció una variable en R, lista, correspondiente a la columna de mensajes del data frame telegram. En Python, se inicializaron variables para traducir los mensajes, que luego se guardaron en un archivo .csv. Con los chats traducidos, se cargaron los archivos en Python y se asignaron a un nuevo data frame llamado `mensajes_traducidos`. Posteriormente, se realizó el análisis de sentimientos utilizando dos métodos distintos: *SentimentIntensityAnalyzer* de *NLTK* y *TextBlob*. Cada método proporcionó una perspectiva única sobre el análisis de sentimientos, permitiendo una evaluación exhaustiva de las conversaciones traducidas. De esta manera, se pudo obtener una visión completa y detallada del tono y la emoción presentes en los mensajes, logrando un análisis integral y preciso.

3. Resultados

No únicamente se realizó el análisis de sentimientos sino que se desarrolló un análisis de características específicas o patrones que se pueden extraer de conversaciones de chat, como las que se muestran a continuación. Realizamos primero el análisis completo para el grupo de Gen.

Cuadro 1: Top 10 más activos

Autor	Mensajes
Venegas	887
Javier Hernández	265
Anthony	144
Corella	118
Ashley Padilla	114
Dixon Montero	112
Juan Pablo M.	98
Adrian	93
Luis Fer Amey	90
Gustavo Amador	68

Cuadro 2: Top 5 más editores

Autor	Editados
Javier Hernández	8
Venegas	5
Adrian	4
Ashley Padilla	3
Gustavo Amador	1

Cuadro 3: Top 5 más stickers

Autor	Mensajes
Venegas	88
Javier Hernández	32
Juan Pablo M.	25
Paula	25
Felix	16

Cuadro 4: Top 5 más imágenes

Autor	Mensajes
Venegas	28
Javier Hernández	16
Sofía Bocker	12
Adrian	12
Dixon Montero	9

Se obtuvieron los usuarios que más sobresalían en algunas de las características propuestas, además se obtuvieron datos del chat más generales, no correspondientes únicamente a un usuario de forma individual, estos datos obtenidos son la racha, de la cual se observa que la mayor racha de días fue de 18 días, desde el 2023-11-26 al 2023-12-13, así como que el día más concurrido es el 2024-05-22, con 308 mensajes.

De forma similar se realiza un estudio de patrones y características generales utilizando como base de datos el chat de Telegram del curso en cuestión.

Cuadro 5: Top 10 más activos

Autor	Mensajes
Potoy Juárez	141
Dixon	10
Gerard	8
Luis Fernando Amey	7
Marco Guardia	6
Sofía Bocker	6
Corella	6
Paula	3
Montserrat Beirute Abarca	2
Santiago Actuariales	2

Cuadro 6: Top 6 más imágenes

Autor	Imágenes
Potoy Juárez	13
Sofía Bocker	3
Montserrat Beirute Abarca	1
Luis Fernando Amey	1
Ashley Padilla	1
Corella	1

Además, de forma análoga al chat de Whatsapp, se encuentran características generales del chat, no ligadas únicamente a un usuario de forma individual, estas corresponden a la racha, de forma que se observa que la mayor racha de días fue de 6, desde el 2024-03-20 al 2024-03-25, así como que el día más concurrido es el 2024-05-04, con 50 mensajes.

Finalmente se agrega tanto el análisis de sentimientos, de ambos chats.

Cuadro 7: Análisis de sentimientos WhatsApp

Autor	Negativo	Neutral	Positivo	Compuesto	Polaridad	Subjetivo	Objetivo
Adrian	0.05	0.80	0.14	0.14	0.04	0.23	0.77
Andrés Montero	0.02	0.79	0.18	0.16	0.05	0.17	0.83
Anthony	0.04	0.85	0.11	0.06	-0.01	0.11	0.89
Ashley Padilla	0.07	0.73	0.19	0.09	0.05	0.19	0.81
Corella	0.03	0.73	0.24	0.16	0.11	0.20	0.80
Dixon Montero	0.07	0.81	0.08	-0.01	0.01	0.10	0.90
Fabian Brenes	0.01	0.83	0.14	0.15	0.05	0.21	0.79
Felix	0.02	0.81	0.14	0.07	0.00	0.05	0.95
Gen	0.05	0.95	0.00	-0.15	0.00	0.03	0.98
Gerard	0.04	0.79	0.14	0.04	0.06	0.15	0.85
Gustavo Amador	0.06	0.78	0.14	0.07	0.04	0.13	0.87
Javier Hernández	0.05	0.82	0.12	0.08	0.04	0.13	0.87
Juan Pablo	0.00	0.71	0.29	0.29	0.00	0.00	1.00
Juan Pablo M.	0.06	0.80	0.13	0.05	-0.01	0.09	0.91
Lau Villacís	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Luis Fer Amey	0.12	0.72	0.16	0.05	0.05	0.20	0.80
Marco Guardia	0.07	0.77	0.16	0.04	0.00	0.17	0.83
Paula	0.02	0.89	0.09	0.04	0.01	0.03	0.97
Santiago	0.01	0.93	0.06	0.08	-0.02	0.07	0.93
Sebas	0.07	0.72	0.21	0.08	0.09	0.19	0.81
Sofía Bocker	0.02	0.90	0.08	0.05	0.04	0.07	0.93
Venegas	0.06	0.82	0.11	0.04	0.01	0.11	0.89
Alejandro Brenes	0.00	1.00	0.00	0.00	0.00	0.03	0.97
Alonso	0.00	0.95	0.05	0.06	0.01	0.15	0.85
Dominick Rodríguez Trejos	0.00	0.89	0.11	0.17	0.00	0.00	1.00
Eyeri	0.00	0.77	0.23	0.25	0.00	0.00	1.00
Heizel Obando Leiva	0.00	0.76	0.24	0.34	0.00	0.00	1.00
Jeremy	0.10	0.65	0.25	0.02	0.01	0.11	0.89
Nay	0.00	0.95	0.05	0.07	0.00	0.00	1.00
Roberto Peralta	0.00	1.00	0.00	0.00	0.00	0.00	1.00
+506 8968 3966	0.00	1.00	0.00	0.00	0.00	0.00	1.00

Directamente de esta tabla podemos extraer la composición de sentimientos de cada usuario, pero vamos a hacer énfasis en los que presentan el máximo en cada área y seguido de las tablas se realiza una explicación específica del significado de cada columna.

El más positivo es Juan Pablo con una valor de 0.2895, el más negativo Luis Fer Amey con 0.1203, la usuario más neutral corresponde a Lau Villacis con un valor de 1, el más subjetivo es Adrian con 0.2263, finalmente el más objetivo es Juan Pablo con un valor de 1.

Cuadro 8: Análisis de Sentimientos Telegram

Autor	Negativo	Neutral	Positivo	Compuesto	Polaridad	Subjetivo	Objetivo
Adrian	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Anthony	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Ashley Padilla	0.00	1.00	0.00	0.00	0.14	0.27	0.73
Corella	0.00	1.00	0.00	0.00	0.04	0.07	0.93
Dixon	0.03	0.74	0.13	0.10	0.08	0.19	0.81
Gerard	0.02	0.85	0.13	0.10	0.01	0.17	0.83
Javier Hernández	0.00	0.54	0.46	0.78	0.88	0.60	0.40
Jose Pablo Trejos Conejo	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Lau Villacís	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Luis Fernando Amey	0.00	0.70	0.30	0.21	0.09	0.12	0.88
Marco Guardia	0.00	0.78	0.22	0.39	0.09	0.45	0.55
Montserrat Beirute Abarca	0.00	0.56	0.44	0.39	0.30	0.30	0.70
Naydelin	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Paula	0.00	0.90	0.10	0.22	0.26	0.35	0.65
Potoy Juárez	0.03	0.84	0.09	0.13	0.07	0.27	0.73
Santiago Actuariales	0.00	1.00	0.00	0.00	0.00	0.00	1.00
Sofía Bocker	0.07	0.92	0.01	0.06	0.00	0.00	1.00
dom	0.00	0.26	0.74	0.44	0.20	0.20	0.80

Directamente de esta tabla podemos extraer la composición de sentimientos de cada usuario, pero vamos a hacer énfasis en los que presentan el máximo en cada área y seguido de las tablas se realiza una explicación específica del significado de cada columna.

El más positivo es dom con un valor de 0.744, la más negativa es Sofía Bocker con 0.0672, el usuario más neutral corresponde a Adrian con un valor de 1, el más subjetivo es Javier Hernandez con 0.6, finalmente el más objetivo es Adrian con un valor de 1.

La columna “negativo” representa la puntuación de negatividad en el texto analizado. Esta métrica se calcula identificando y sumando términos o frases con carga emocional negativa dentro del texto. Normalmente, los valores varían entre 0 y 1, donde 0 indica la ausencia de términos negativos y 1 representa un texto altamente negativo. Un valor más alto en esta columna indica una mayor presencia de palabras con connotaciones negativas, tales como "triste", "malo", "horrible", etc. Esta métrica es especialmente útil para identificar textos que pueden necesitar atención debido a su naturaleza potencialmente perjudicial o conflictiva.

La columna "neutral" indica la puntuación de neutralidad del texto. Representa la cantidad de contenido que no expresa una fuerte carga emocional, ni positiva ni negativa. Al igual que la columna de negatividad, los valores suelen estar entre 0 y 1. Valores más altos sugieren que el texto es principalmente neutral, con una predominancia de términos descriptivos o informativos que no evocan emociones intensas. Esta métrica es útil para determinar si un texto es mayormente informativo o descriptivo sin un fuerte sesgo emocional, como reportes de noticias, documentos técnicos, o artículos enciclopédicos.

La columna "positivo" muestra la puntuación de positividad del texto. Se calcula mediante la identificación y suma de términos o frases con connotaciones positivas. Normalmente, los valores varían entre 0 y 1. Un valor

más alto indica una mayor presencia de palabras con connotaciones positivas, como 'feliz', 'excelente', 'maravilloso', etc. Esta métrica es valiosa para identificar textos que son alentadores, motivadores o que contienen feedback positivo.

La columna “compuesto” representa la puntuación compuesta del análisis de sentimientos, integrando las puntuaciones de negatividad, neutralidad y positividad para proporcionar una visión global del sentimiento del texto. Los valores pueden variar más ampliamente, a menudo entre -1 y 1. Valores positivos indican un sentimiento general positivo, valores negativos indican un sentimiento general negativo, y valores cercanos a 0 indican un equilibrio entre elementos positivos y negativos o una predominancia de elementos neutrales. Esta métrica ofrece una visión rápida y consolidada del sentimiento general de un texto, útil para resúmenes ejecutivos o análisis de grandes volúmenes de texto.

La columna “polaridad” mide la dirección y el grado de emoción en el texto, con un enfoque en la diferenciación entre sentimientos positivos y negativos. Los valores típicamente van de -1 a 1. Un valor de -1 indica una fuerte negatividad, mientras que un valor de 1 indica una fuerte positividad, y 0 indica neutralidad. Valores cercanos a -1 sugieren un fuerte sentimiento negativo, valores cercanos a 1 sugieren un fuerte sentimiento positivo, y valores cercanos a 0 sugieren un sentimiento neutral. Esta métrica es útil para obtener una visión rápida del tono general del texto, especialmente en análisis comparativos.

La columna “subjetividad” mide cuán subjetivo es el texto, indicando la cantidad de opinión versus hecho presente en el contenido. Los valores generalmente varían entre 0 y 1. Un valor de 0 indica un texto completamente objetivo, mientras que un valor de 1 indica un texto completamente subjetivo. Valores altos cercanos a 1 indican que el texto está basado en opiniones personales, percepciones o juicios, mientras que valores bajos cercanos a 0 indican que el texto está basado en hechos verificables y objetivos. Esta métrica es útil para distinguir entre textos que proporcionan información objetiva, como artículos científicos, y aquellos que expresan opiniones, como editoriales o comentarios de blogs.

Además se obtuvo la racha, que corresponde a la mayor cantidad de días consecutivos que los chat estuvieron activos, así como el promedio de las horas con mayor actividad, y el día específico que presente la mayor cantidad de actividad así como la cantidad de mensajes.

4. Conclusiones

El análisis realizado revela patrones significativos y características específicas en las conversaciones de chat tanto en WhatsApp como en Telegram. A través del análisis de sentimientos, se pudo identificar las emociones predominantes y el tono de los mensajes tanto de forma general como analizando individuos de forma específica, proporcionando una visión integral del estado de ánimo y la interacción en los grupos y sus integrantes.

Además, los análisis de actividad destacaron los usuarios más activos y sus contribuciones específicas, permitiendo identificar a los participantes clave y sus roles dentro de la dinámica grupal. La identificación tanto de la racha de días activos como de los picos de actividad diaria ofrecen una comprensión detallada del intervalo temporal y la intensidad de la actividad en los grupos de chat, siendo estas características de utilidad, para llevar a cabo anuncios o estudio de la interacción entre los usuarios.

En conjunto, estos análisis proporcionan una herramienta adecuada para el estudio y gestionamiento de una mejor comunicación dentro de grupos, facilitando la identificación de tendencias, la resolución de conflictos y la mejora de las distintas interacciones que se desarrollan de forma grupal.

5. Recomendaciones y Extensión de Resultados

A partir de los análisis realizados, se pueden ofrecer varias recomendaciones. Para mejorar la participación y unión del grupo, resulta fundamental fomentar la participación activa, incentivando a los usuarios menos activos a que se integren más a las conversaciones que se desarrollan, esto con el fin de crear un ambiente inclusivo y amigable para que nuevas personas puedan expresarse. Además, implementar mecanismos de reconocimiento para los usuarios más activos y aquellos que aportan contenido valioso que pueda presentar beneficios para los demás. El manejo de conflictos es otra área clave; utilizar los análisis de sentimientos para identificar y abordar posibles conflictos antes de que escalen puede mantener la paz y tranquilidad en el grupo, no afectando a los usuarios que no se ven envueltos en el conflicto.

En cuanto a la optimización de la comunicación, coordinar las actividades del grupo con respecto a los horarios de mayor actividad con la intención de maximizar la participación de los distintos miembros. Generando de esta forma un espacio adecuado para los mensajes con una naturaleza informativa, para mantener los chats con un grado de utilidad para los respectivos miembros.

Para extender el análisis, sería valioso ampliar los datos a otros periodos de tiempo y otros grupos para poder realizar un estudio más amplio acerca de como se desarrollan las dinámicas de comunicación con miembros distintos. Integrar datos de otros medios de comunicación utilizados por los miembros del grupo, como correos electrónicos o reuniones, puede proporcionar un análisis mucho más completo. Además, brinda la posibilidad de desarrollar modelos predictivos para anticipar tendencias de comportamiento de los usuarios, así como una predicción de las necesidades que se desarrollen dentro del grupo.

La implementación de herramientas tecnológicas puede resultar provechosa para la gestión adecuada del grupo. La implementación de bots de moderación que utilicen análisis de sentimientos en tiempo real para moderar el chat y buscar mantener un ambiente positivo. Finalmente, automatizar un proceso de generación de reportes de forma periódica de actividad y sentimiento permitiría llevar a cabo un análisis específico en intervalos de tiempo más reducidos.

6. Anexos

Para el seguimiento del proyecto, se le indica al lector que visite el repositorio de **GitHub**, o en el siguiente link: <https://github.com/bluke7ide/Proyecto-Herramientas>

Referencias

- Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
- Blogathon, D. S. (2022). Making natural language processing easy with textblob. <https://www.analyticsvidhya.com/blog/2021/10/making-natural-language-processing-easy-with-textblob/>. Fecha de Acceso: 01 de Mayo de 2024.
- Grolemund, G. and Wickham, H. (2023). *lubridate: Make Dealing with Dates a Little Easier*. R package version 1.9.2.
- Horn, J. (2023). Análisis de textos en python: de lo básico al aprendizaje profundo. <https://konfuzio.com/es/analisis-de-texto-en-python/#text-analysis-in-python-â–grundlagen>. Fecha de Acceso: 01 de Mayo de 2024.
- Jablonski, J. (S.F.). Natural language processing with python's nltk package. <https://realpython.com/nltk-nlp-python/>. Fecha de Acceso: 01 de Mayo de 2024.
- Loria, S. (2018). *TextBlob: Simplified Text Processing*.
- Mayo, M. (2018). Comenzando con spacy para procesamiento de lenguaje natural. <https://medium.com/datos-y-ciencia/comenzando-con-spacy-para-procesamiento-de-lenguaje-natural-e8cf24a18a5a>. Fecha de Acceso: 01 de Mayo de 2024.
- Narvaez, M. (S.F.). ¿qué es el análisis de texto? <https://www.questionpro.com/blog/es/analisis-de-texto/>. Fecha de Acceso: 01 de Mayo de 2024.
- Ooms, J. (2023). *jsonlite: A Simple and Robust JSON Parser and Generator for R*. R package version 1.8.4.
- Rodrigo, J. A. (2020). Análisis de texto (text mining) con python. <https://cienciadedatos.net/documentos/py25-text-mining-python>. Fecha de Acceso: 01 de Mayo de 2024.
- Ushey, K., Allaire, J., Tang, Y., and Stephens, J. (2023). *reticulate: R Interface to Python*. R package version 1.30.
- Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.
- Wickham, H. (2023). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.3.2.
- Wickham, H., Hester, J., and François, R. (2023). *readr: Read Rectangular Text Data*. R package version 2.1.4.
- Wolf, T. and Debut, L. (2020). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Zidan, I. (2020). *Deep-translator: A flexible free translation tool in Python*.