Polo del Conocimiento

Pol. Con. (Edición núm. 63) Vol. 6, No 11
Noviembre 2021, pp. 104-123
ISSN: 2550 - 682X
DOI: 10.23857/pc.v6i11.3257

Métodos de extracción de comentarios de la red social Twitter para uso en Procesamiento de Lenguaje Natural

Methods for extracting comments from the social network Twitter for use in Natural Language Processing.

Métodos de extração de comentários da rede social Twitter para uso no Processamento de Linguagem Natural

César Espin Riofrio [I]
cesar.espinr@ug.edu.ec
https://orcid.org/0000-0001-8864-756X

Angélica Cruz Chóez [II]
angelica.cruzc@ug.edu.ec
https://orcid.org/0000-0003-4711-6820

Johanna Zumba Gamboa [III]
johanna.zumbag@ug.edu.ec
https://orcid.org/0000-0001-5733-5678

Correspondencia: cesar.espinr@ug.edu.ec

Ciencias Técnicas y Aplicadas
Artículo de investigación

Recibido:** 30 de Septiembre de 2020 ***Aceptado:** 22 de Octubre de 2021 * Publicado:** 09 de Noviembre de 2021

I. Magister en Sistemas de Información Gerencial, Universidad de Guayaquil, Ecuador.
II. Magister en Sistemas Integrados de Gestión, Universidad de Guayaquil, Ecuador.
III. Magister en Sistemas de Información Empresarial, Universidad de Guayaquil, Ecuador.

Resumen

En la actualidad, el avance tecnológico en el desarrollo de herramientas para la extracción de comentarios de las redes sociales y páginas web está dando pasos agigantados, la evolución de la web ha obligado a todas las organizaciones a adaptarse a este fenómeno digital. Las grandes corporaciones han desarrollado e implementado sus propias APIs en las plataformas de mayor concurrencia de usuarios en redes sociales como lo son Facebook, YouTube, WhatsApp, Instagram, Twitter y también la creación de herramientas web scraping para plataformas que no cuentan con estas. Es de mucha importancia contar con un corpus de datos que permitan ser analizados en tareas relacionadas al Procesamiento de Lenguaje Natural. Como objetivo del presente trabajo se realizan pruebas de herramientas de extracción de comentarios de la red social Twitter. El método utilizado para extraer comentarios de Twitter es a través de las APIs de tipo Rest y Streaming y una herramienta web scraping. Estos comentarios son indexados y para ser enviados a una base de datos no relacional que maneja grandes volúmenes de información. Como resultado se verifica la capacidad de extracción de las herramientas creando un corpus con comentarios de Twitter para análisis utilizando técnicas de Procesamiento de Lenguaje Natural (PLN). Se concluye que es factible la extracción de tweets mediante herramientas diseñadas para el efecto o utilizando librerías libres en Python para el efecto.

Palabras Clave: Procesamiento de Lenguaje Natural; Twitter; web scraping; corpus.

Abstract

Currently, technological progress in the development of tools for extracting comments from social networks and web pages is taking giant steps, the evolution of the web has forced all organizations to adapt to this digital phenomenon. Large corporations have developed and implemented their own APIs on the platforms with the highest number of users on social networks such as Facebook, YouTube, WhatsApp, Instagram, Twitter, and the creation of web scraping tools for platforms that do not have them. It is very important to have a corpus of data that can be analyzed in tasks related to Natural Language Processing. The objective of this work is to test tools for extracting comments from the social network Twitter. The method used to extract comments from Twitter is through Rest and Streaming APIs and a web scraping tool. These comments are indexed and to be sent to a non-relational database that handles large volumes of

information. As a result, the extraction capacity of the tools is verified by creating a corpus with Twitter comments for analysis using Natural Language Processing (NLP) techniques. It is concluded that it is feasible to extract tweets using tools designed for this purpose or using free Python libraries for this purpose.

Keywords: Natural Language Processing; Twitter; web scraping; corpus.

Resumo

Atualmente, o avanço tecnológico no desenvolvimento de ferramentas para a extração de comentários de redes sociais e páginas da web dá passos gigantes, a evolução da web obrigou todas as organizações a se adaptarem a este fenômeno digital. Grandes corporações desenvolveram e implementaram suas próprias APIs nas plataformas com maior número de usuários em redes sociais como Facebook, YouTube, WhatsApp, Instagram, Twitter e também a criação de ferramentas de web scraping para plataformas que não as possuem. É muito importante ter um corpus de dados que possa ser analisado em tarefas relacionadas ao Processamento de Linguagem Natural. Como objetivo deste trabalho, são realizados testes de ferramentas de extração de comentários da rede social Twitter. O método usado para extrair comentários do Twitter é por meio de APIs Rest e Streaming e uma ferramenta de web scraping. Esses comentários são indexados e enviados a um banco de dados não relacional que lida com grandes volumes de informações. Como resultado, a capacidade de extração das ferramentas é verificada através da criação de um corpus com comentários do Twitter para análise usando técnicas de Processamento de Linguagem Natural (PNL). Conclui-se que é possível extrair tweets usando ferramentas projetadas para esse fim ou usando bibliotecas Python gratuitas para esse fim.

Palavras-chave: Processamento de Linguagem Natural; Twitter; Raspagem da web; corpus.

Introducción

Las tecnologías web a lo largo del tiempo han evolucionado de una manera acelerada, paralelo a esto las redes sociales, páginas web y aplicaciones que interconectadas entre si generan información de diferentes tipos como textos, audios, videos, imágenes y contenido interactivo a gran escala. Esto obligó al área informática y tecnológica a crear y desarrollar herramientas para

la extracción, almacenamiento y tratamiento de datos con el fin de sacar provecho de todo este gran flujo de información.

La red social Twitter que desde su aparición en el año 2006 resultó ser un fenómeno social, siendo un punto clave para la comunidad científica debido a sus reducidos mensajes de 280 caracteres. Esta red social es una tendencia que ha transformado los textos cortos en pieza clave para la comunicación entre la sociedad, Twitter se involucra también en estudios como la clasificación de textos (Schulz et al., 2014), el Análisis supervisado de sentimientos en Twitter utilizando skipgrams (Fernández et al., 2015), y el Análisis de sentimientos en Twitter para el español (Pla & Hurtado, 2014), a esto también se suma la geolocalización (Han et al., 2014), y un sin números de estudios que pretenden extraer datos desde esta plataforma digital.

Twitter es la plataforma de comunicación en tiempo real más utilizada del mundo en internet, se alimenta de millones de usuarios que interactúan con sus opiniones, inquietudes e información, estas actividades generan gran cantidad de datos en su mayor parte de tipo público, pero lo que hace atractiva a esta red social es su sencillez, su estructura y los objetos que posee con sus posibles relaciones como muestra la figura 1, esto despierta el interés de las grandes comunidades científicas, empresariales y demás sectores.

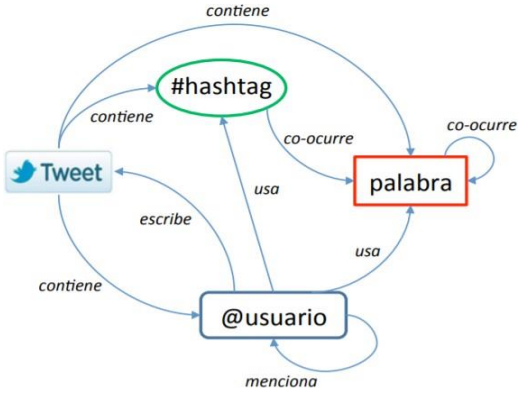

Figura 1 Objetos de Twitter con sus posibles relaciones

Twitter es considerado un canal de comunicación que ha sido implementado por las personas, instituciones y organismos a nivel mundial. Según (Bernhardt et al., 2014), Es una de las plataformas sociales con mayor valor para el desarrollo profesional. La tendencia que se aprecia es que los seguidores o followers han ido creciendo progresivamente. Se estima que esta red

social cuenta con 340 millones de usuarios activos aproximadamente, esto hace que se genere 500 millones de tweets por día.

La creación de un corpus es una tarea necesaria para las técnicas de Procesamiento de Lenguaje Natural (PLN), Minería de Opiniones (MO) y otros campos de investigación como la lingüística y los diferentes tipos de análisis. El proceso de recopilación y creación de un corpus suele llevar mucho tiempo si se realiza manualmente, por ejemplo, descargando textos de la web o el uso de software adicional para la compilación automática de la información (Fantinuoli, 2016).

Una de las fuentes principales para la elaboración de corpus son las opiniones de la red social Twitter, esto es debido al gran volumen de información que se genera, lo interesante es que abarca temas de aspectos sociales derivados de su entorno y característico rasgo divulgativo de informar (Martínez-Cámara et al., 2014).

Un corpus es una colección extensa de datos tipo texto ya sea escrito u oral en formato electrónico, de miles y millones de palabras que se codifican y se clasifican de manera adecuada, estos son procesados y almacenados en medios masivos, son creados con el propósito de realizar diferentes tipos de búsquedas entre grandes cantidades de textos digitales (Pitkowski & Vásquez Gamarra, 2009).

Existen corpus en el idioma español, uno de ellos es el perteneciente al Taller de Análisis de Sentimientos (TASS) de la Sociedad Española para el Procesamiento de Lenguaje Natural (SEPLN) , donde esta sociedad tiene como objetivo la promoción de la investigación en la rama del Procesamiento de Lenguaje Natural (PLN) y de las Tecnologías del Lenguaje Humano (TLH), esta sociedad organiza cada año una competición donde se presentan distintos métodos para la extracción y clasificación de tweets, todo esto empezó desde el año 2012 y su novena convocatoria fue en el año 2020. Cabe destacar que, aunque La SEPLN no publique un corpus nuevo por cada año, existen 3 que se los utilizaron en estudios previos para entrenamiento y evaluación de clasificadores, se detalla a continuación sus características:

General Corpus, contiene 68000 tweets en el idioma español de 150 personas conocidas dentro del campo de la política, economía, cultura y comunicación, los tweets fueron extraídos entre 11/2011 y 03/2012. (Villena-Román & García-Morera, 2013).

Politics Corpus, es un corpus de tipo específico, se lo consiguió durante elecciones de las Cortes Generales de España en 2011 y cuenta con 2500 tweets, estos contienen información de los 4

relevantes partidos de ese entonces, IU, PSOE, PP y UPyD. (Villena-Román & García-Morera, 2013)

International TASS Corpus (InterTASS), es un corpus de tipo general que reúne 3400 tweets en el idioma español, su contenido abarca sobre cualquier tema en general. International TASS Corpus (InterTASS) es un corpus publicado en 2017 (Manuel C. Díaz-Galiano et al., 2018) que fue actualizado en 2018 (Manuel Carlos Díaz-Galiano et al., 2019)

Otro trabajo interesante es el Corpus de Aprendices de Español como Lengua Extranjera (CAES) (Mrtinez, 2015). Este corpus pertenece a un conjunto de textos escritos realizado por estudiantes de español como Espanhol como Língua Estrangeira (ELE), en todos los niveles de competencia. Su elaboración consta de diferentes lenguas maternas como lo son: árabe, chino, mandarín, francés, inglés, portugués y ruso, cuenta con un total de 3878 textos, este corpus fue elaborado por docentes e investigadores de la Universidad de Santiago de Compostela y del Instituto de Cervantes, en España.

Todas las fuentes que fueron encontradas, revisadas y analizadas coinciden con la necesidad de crear herramientas dirigidas al proceso de extracción de datos, además de la creación de un corpus aprovechando la fuente inagotable de datos generados en las redes sociales y plataformas virtuales, para este caso específico Twitter. Toda la información antes mencionada se puede extraer con las herramientas de extracción enfocadas al PLN. También se manejaron las APIs de Twitter, las herramientas Web Scraping y crawling, que no solo extraen información, sino que también realizan análisis y estadísticas que buscan conocer ciertas entidades.

El Procesamiento de Lenguaje Natural es una rama de las ciencias de la computación, Inteligencia Artificial, la lingüística que se encarga del estudio de las interacciones que tiene el lenguaje humano con las computadoras, todo esto por medio de los análisis sintáctico, semántico, morfológico y pragmático. Se basa en reglas de patrones estructurales empleando el formalismo gramatical concreto. Estos patrones se definan al momento de combinar las reglas con toda la información almacenada en diccionarios de computación, con el fin de realizar reconocimientos de letras u oraciones, también por medio de imágenes, texto o voz (Instituto de Ingeniería Del Conocimiento, . Procesamiento Del Lenguaje Natural ¿qué Es? Recuperado de Http://Www.Iic.Uam.Es/Inteligencia/Que-Es-Procesamiento-Del-Lenguaje-Natural/, 2017).

En el presente artículo se propone aportar con soluciones para extraer tweets con el propósito de crear un corpus. Como primera opción se plantea utilizar Twitter con sus respectivas APIs, las

mismas que ayudarán a extraer información ya sea en tiempo real streaming (Streaming, 2021) o con datos recientes Rest Api ("REST API.," 2021), en conjunto con las librerías Tweepy y Twint de Python. En la segunda opción se utiliza una herramienta web scraping llamada Octoparse con el mismo propósito de extraer comentarios. Toda esta información obtenida se almacenará en una base de datos no relacional, se emplea Elasticsearch con la finalidad de crear un corpus, por último, se procede a realizar consultas básicas mediante el framework Kibana, componente del paquete Elasticsearch.

Metodología

Se utiliza la metodología documental, se revisan artículos científicos y libros a través de fuentes bibliográficas, debido a la amplitud que este tema abarca. También se emplea la metodología experimental que aborda los métodos de extracción de opiniones de Twitter para la creación de un corpus, es decir, esto conlleva al estudio de las APIs y soluciones ofrecidas por Twitter.

Parte de esta investigación es configurar las APIs de Twitter con un algoritmo en el lenguaje de programación Python y sus librerías Tweepy y Twint que permite definir en sus métodos filtros de búsquedas y devuelve información dentro de objetos en formatos Json que se almacenan en la base de datos.

Se procede también a extraer las opiniones con la herramienta de tipo web scraping Octoparse y los datos se guardarán en formato csv. Para almacenar toda esta información se decide trabajar con la base de datos NoSQL Elasticsearch, por poseer un excelente motor de búsqueda, además de ser flexible y potente, posee buena escalabilidad en volumen de datos en tiempo real, es de código abierto y distribuido, para finalizar se realizará consultas con el framework frontend Kibana.

La arquitectura aplicada es llamar a Twitter a través de su API, extraer los comentarios con las bibliotecas de Python Tweepy y Twint y también el uso de la herramienta Octoparse para luego almacenar los comentarios en la base no relacional Elasticsearch, una vez ahí pueden ser visualizados utilizando Kibana.

La elaboración y ejecución del modelo propuesto en la figura 3 se basa en un proceso de fases importantes para la realización de este, tomando como referencias los antecedentes de estudios

previos de investigaciones realizadas, garantizando una continuación de proyectos enfocados a la extracción de opiniones con el propósito de crear corpus de información.

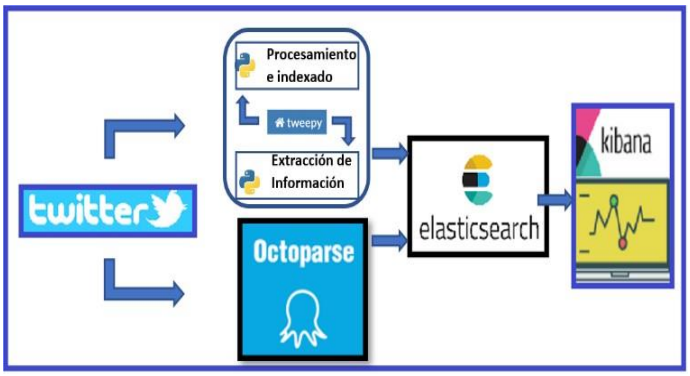

Figura 2 Arquitectura aplicada al trabajo de investigación

En el presente trabajo se procede a extraer información tanto con la herramienta web scraping y las APIs de Twitter en sus modalidades de Rest y Streaming, se debe de conseguir permisos de esta red social, para obtener el extenso contenido de datos que se genera, es imprescindible obtener sus credenciales correspondientes de acceso. Se toma como lenguaje de programación a Python para crear un algoritmo que nos permita extraer toda esta información con sus librerías Tweepy y Twint, así mismo se procede a enviar estos datos a una base de datos NoSQL Elasticsearch, con el fin de crear un corpus.

Web scraping es el proceso de recolectar datos contenidos en páginas web mediante técnicas automatizadas. Lo distintivo del web scraping es que en principio los datos parecen poco estructurados. Corresponde por tanto al analista de datos identificar cuál es el patrón que siguen los datos, para luego crear y ejecutar un algoritmo de extracción y procesamiento de estos. Las APIs de Twitter presentan limitaciones como muestra la tabla 1.

Tabla 1 APIs de Twitter y sus limitaciones

API	Petición	Máximo de datos	Cada 15 Minutos
REST	GET user_timeline	200 tweets	900*200=180.000 tweets
Streaming	POST statuses_filter	-	Máximo 45.000 tweets
Search	GET search/tweets	100 tweets	180*100 = 18.000 tweets

Extracción de tweets mediante Octoparse

Para el uso de esta herramienta es necesario un registro en la página oficial y activación de la cuenta, una vez realizada estas acciones se puede iniciar sesión a la herramienta. En la figura 3 se muestra el tablero del aplicativo una vez iniciada la sesión y donde se podrá hacer uso de esta herramienta para extraer datos en base a una página web (dirección URL), en la prueba realizada se utiliza la página oficial de una cadena televisiva.

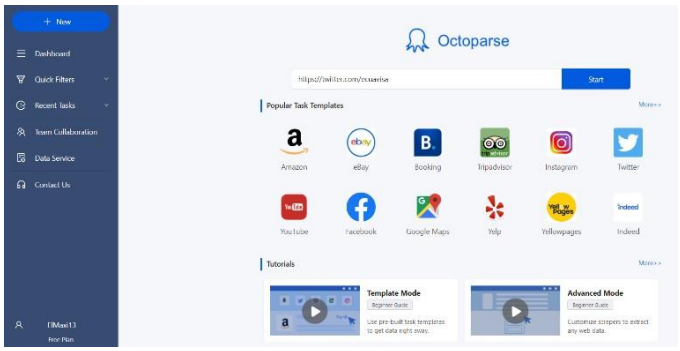

Figura 3 Herramienta de extracción Web Scraping Octoparse

Se ingresa la dirección web la manera directa, es decir, se insertar la dirección URL en el cuadro te texto y se visualiza la página a la cual se desea extraer los datos, figura 4. Es necesario crear una nueva tarea en donde se hace el ingreso de la URL.

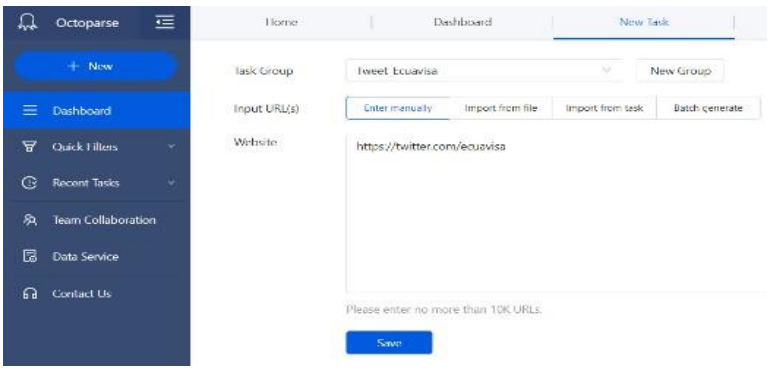

Figura 4 Ingreso de URL en nueva tarea

Pol. Con. (Edición núm. 63) Vol. 6, No 11, Noviembre 2021, pp. 104-123, ISSN: 2550 - 682X

La herramienta cuenta con dos métodos de extracción de datos de una página web; automática, la propia herramienta verifica la estructura de la página web y extrae los datos y manual, el usuario decide qué datos desea extraer.

Para el análisis de esta herramienta de extracción de datos se emplea el método de selección manual, debido a que sólo se requiere extraer campos como: usuario, tweets, fecha de publicación del tweet, entre otros.

Twitter emplea "desplazamiento infinito", significa que primero se tiene que ir desplazando hacia abajo toda la página para lograr que Twitter permita cargar la mayor cantidad de tweets, y luego proceder a extraer la información. La estructura de la página de Twitter facilita al momento de seleccionar los contenedores, es decir, con seleccionar los dos primeros contenedores la herramienta interpreta la estructura de los siguientes seleccionando de manera automática como se puede observar en la figura 5.

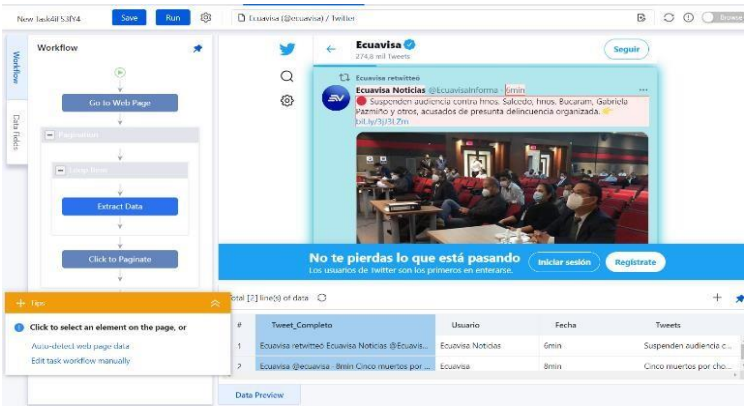

Figura 5 Selección de campos a extraer

Se necesita que Twitter cargue el contenido antes de extraer, para ello se configuran los campos click to paginate y pagination para el tiempo de espera para que Twitter cargue el contenido y el ciclo que termina después de un número determinado de iteraciones, respectivamente.

Realizada las configuraciones se procederá a la ejecución y la herramienta empezará con la extracción de los tweets, para efectos de pruebas se realizó un número 300 iteraciones, obteniendo un archivo .csv. En el proceso de extracción se pueden encontrar datos duplicados los cuales son eliminados, esta herramienta ofrece la opción de eliminarlos antes de almacenar.

La herramienta frontend Kibana mantiene una estrecha integración con Elasticsearch y proporciona herramientas para ciertas actividades como visualización, indexación de

documentos, búsqueda de datos, etc. En la página de inicio de Kibana se puede importar archivos y almacenarlos en un índice, se lleva a cabo el almacenamiento del archivo .csv obtenido luego de la extracción, se aprecia la importación en la imagen 6.

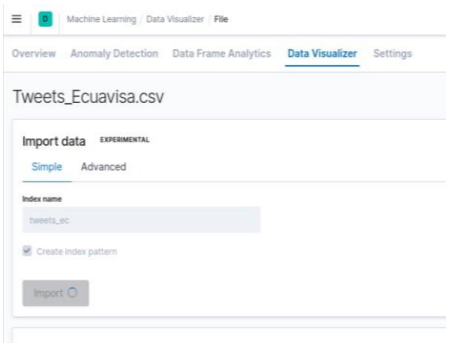

Figura 6 Importación de archivo .csv

Extracción de tweets mediante las API de Twitter

Para poder hacer uso de la API de Twitter, debe estar registrado en la red social para obtener claves de autenticación que se muestran en la figura 7, las mismas que se obtienen en el apartado de desarrollador (dev Twitter) creando una app de Twitter.

Figura 7 Claves de autenticación

En la presente prueba se llevan a cabo el desarrollo de 2 algoritmos con las APIs de Twitter, el primer algoritmo (Rest) recolecta tweets publicados recientemente, haciendo un recorrido desde el actual. En el segundo algoritmo (Streaming) se obtendrán los tweets que se están publicando en tiempo real.

Las credenciales de autenticación se ingresan en un archivo Python adicional, el mismo que guardará estos datos en un archivo tipo JSON con la finalidad de abrir el archivo en los

distintos algoritmos que requieran consumir las APIs de Twitter como se puede visualizar en la figura 8.

```
with open("twitter_credentials.json", "r") as file:
    creds = json.load(file)
```

Figura 8 Lectura del archivo JSON con las claves

El estándar abierto Open Authorization (Oauth) permite a las aplicaciones del cliente una autenticación segura a una API, para esto se usa la librería Tweepy que permite esa autenticación, en a figura 9 se aprecia este procedimiento. De esta manera se accede a la API y se procede a realizar la extracción de tweets para crear un corpus con estos datos.

```
# autenticamos para poder hacer uso de la api de twitter
auth = OAuthHandler(creds['CONSUMER_KEY'], creds['CONSUMER_SECRET'])
auth.set_access_token(creds['ACCESS_TOKEN'], creds['ACCESS_SECRET'])

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

Figura 9 Autenticación y uso de la API de Twitter

API Rest

Para poder manejar grandes cantidades de datos en Tweepy se hace uso de *cursor*, el mismo que se lo irá recorriendo dentro de un ciclo repetitivo, esto con la finalidad de recorrer todos los tweets debido a que Tweepy maneja paginaciones, y por cada página sólo se obtendría 20 datos, de ahí la importancia de utilizar un cursor, dentro del mismo se configura el criterio de búsqueda.

El método de búsqueda search devuelve los tweets relevantes que coincidan con un criterio de búsqueda, en la figura 10 dentro del cursor se coloca este método, y se usa la localización (latitud, longitud y radio), el parámetro (q) es una cadena de consulta, puede ser una palabra o frase que no exceda a los 500 caracteres. El objeto que devuelve el cursor es un compendio de datos por cada tweet en formato JSON, de los cuales solo requiere obtener el usuario, fecha de creación del tweet y el contenido del tweet.

```
count = 0
for status in tweepy.Cursor(api.search, q='efrain ruales', lang="es",
                            loc="-79.8868702,-2.1900703,50 km,",
                            tweet_mode="extended").items(2000):
    count +=1
    if 'retweeted_status' in status._json:
        estado=status._json['retweeted_status']['full_text']
    else:
        estado=status.full_text

    print(F"Usuario: {status._json['user']['screen_name']}")
    usuario = status._json['user']['screen_name']
    print(f"Fecha de publicacion: {status._json['created_at']}")
    fecha = status._json['created_at']
    print(f"Estado {count}: {estado} \n")
```

Figura 10 Ciclo repetitivo para Cursor

Los datos son almacenados en Elasticsearch gracias a la librería de cliente que ofrece, como en todo manejo de bases de datos se requiere establecer conexión y posterior envío a almacenamiento de los datos, en este caso el almacenamiento a Elasticsearch se realiza de manera local, el puerto por defecto es el 9200, es=Elasticsearch([{'host':'localhost','port':9200}])

Elasticsearch almacena documentos en formato JSON, por lo tanto, se preparan los datos en dicho formato con el método index, donde se especifican los datos indispensables como el nombre del índice, el tipo de documento y el cuerpo para detallar los campos que se pretenden almacenar en el índice, cada documento tiene un identificador para el presente algoritmo pero no lo especifica, por lo que Elasticsearch asigna uno de manera automática, figura 11, cabe mencionar que se debe levantar los servicios tanto de Elasticsearch como los de Kibana antes de la ejecución del archivo.

```
es.index(index='tweets_REST',
         doc_type='comentarios',
         body={
             'usuario': usuario,
             'fecha':fecha,
             'twit': estado,
         })
```

Figura 11 Indexación de los datos a Elasticsearch

Por consola se muestran todos los datos que se van obteniendo, figura 12.

Figura 12 Datos obtenidos

Para un correcto proceso de creación del corpus, estos datos extraídos fueron indexados en documentos tipo JSON, este tipo de datos son compatibles con Elasticsearch, esto con el propósito de poder consultar y visualizar toda la información de los tweets, para esto se utilizó el framework Kibana.

API Streaming

Para el proceso de extracción de tweets en tiempo real se crea una clase, la misma que hereda de StreamListener con tres métodos para establecer conexión, controlar errores, y extraer datos, en este último es donde se establece los índices para Elasticsearch.

```
class TweetsListener(tweepy.StreamListener):
    def on_connect(self):
        print("Coneccion exitosa...")

    def on_status(self, status):
        print("Estado:  ")
        print(status._json["user"]["screen_name"])
        usuario = status._json["user"]["screen_name"]
        print(status._json["created_at"])
        fecha = status._json["created_at"]
        print(status._json["text"])
        estado=status._json["text"]

        es.index(index='tweets_STREAMING',
                 doc_type='comentarios',
                 body={
                     'usuario': usuario,
                     'fecha': fecha,
                     'twit': estado,
                 })


    def on_error(self, status_code):
        print("Error", status_code)
```

Figura 13 Clase para el Streaming

Una vez realizada la autenticación (figura 9) y si el estado escucha a los estados, se puede crear un objeto de estudio, es decir, un objeto de la clase creada por medio del constructor y Tweepy, se pasa como parámetro la autenticación y el objeto, posteriormente se realiza la transmisión, los datos que se requieren serán filtrados, en este caso por localidad como se observa en la figura 14.

```
stream = TweetsListener()
streamingApi = tweepy.Stream(auth=api.auth, listener=stream)
streamingApi.filter(locations=[-79.9577820301,-2.2883024357,-79.8465454578,-2.0948101259])
```

Figura 14 Instancia de la clase y filtro de los tweets

Para la localidad se utiliza un cuadro delimitador con la herramienta Bounding Box, la misma que ofrece una interfaz amigable como se visualiza en la figura 15.

Figura 15 Cuadro delimitador geolocalizador, los puntos requeridos son de tipo CSV RAW

La ejecución en este método también se visualiza por consola como se observa en la siguiente figura 16.

Figura 16 Ejecución del algoritmo Streaming

Discusión

La extracción de datos de la red social Twitter se ha podido realizar utilizando varios métodos según se evidencia en este trabajo, mediante el uso de herramientas comerciales o de bibliotecas en Python específicas para el efecto, en este último caso dichas librerías brindan varias opciones para filtrar la extracción de tweets.

La recopilación de datos es de suma importancia para la formación de un corpus con el cual se pueden hacer análisis en distintas tareas del Procesamiento del Lenguaje Natural como el análisis de sentimiento, clasificación de tópicos, detección de noticias falsas, etc.

El almacenamiento y administración del corpus de datos recopilado puede realizarse en bases de datos no estructuradas como Elasticsearch usada en el presente trabajo que cuenta con Kibana para las consultas y visualización de estos. Elasticsearch cuenta con la interfaz para Twitter pudiendo enviar los datos extraídos directamente a la base.

Se ha podido comprobar el uso y utilidad de las herramientas cuyas únicas limitaciones son la cantidad de extracciones y tiempo de búsqueda que permite Twitter. Se podrían realizar extracciones de otras redes sociales como Instagram, Facebook, etc. A través de las APIs que disponen o mediante procesos de web scraping que sirve no solo para redes sociales sino para cualquier sitio web del cual se desee extraer información.

Resultados

La mayor parte de soluciones desarrolladas para extraer tweets se inclinan por usar las APIs de Twitter, estas se encuentran dentro del paquete de funciones destinados para el análisis de las redes sociales, en otros casos se aplican métodos personalizados como propósito de desarrollar herramientas de tipo privadas para una línea investigativa específica enfocada al Procesamiento de Lenguaje Natural (PLN).

Algunas herramientas comerciales que se pueden emplear en este tipo de trabajos son DiscoverText, NodeXL, Octoparse, entre otras.

Con la herramienta Octoparse se obtuvo como resultado la extracción de 596 tweets en un tiempo de 57 min 54 seg como se observa en la figura 17, este resultado será almacenado en un archivo con formato .csv, además cabe mencionar que esta herramienta ofrece otros formatos para poder almacenar estos datos extraídos como Excel, SqlServer, MySql y HTML.

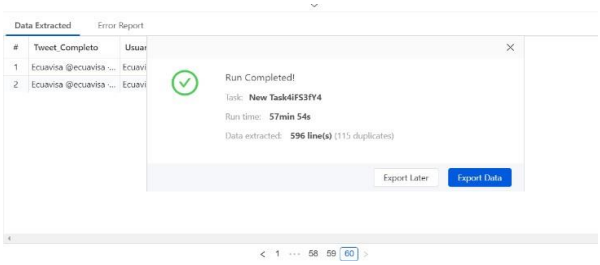

Figura 17 Ventana de aviso de extracción de tweets completa

Al momento de visualizar los datos se realiza consulta con palabras o frases como se observa en la siguiente imagen 18.

Figura 18 Consulta a los tweets extraídos por Octoparse

Pol. Con. (Edición núm. 63) Vol. 6, No 11, Noviembre 2021, pp. 104-123, ISSN: 2550 - 682X

Tanto con la API Rest y la API Streaming se pudieron extraer de forma correcta tweets de acuerdo con los parámetros y filtros indicados, todos esos tweets se los puede llevar a un archivo .csv para luego importarlos en Kibana. Es importante mencionar que los datos extraídos por los dos métodos la hora dependerá de la zona donde se encuentre. En la siguiente tabla se muestra la cantidad de tweets extraídos según el método utilizado.

Tabla 2 Comparativo de datos extraídos con Octoparse y las APIs de Twitter

Descripción	N° de Tweets	Tiempo
Octoparse	596	57.54 min
Api Rest	2000	2.45 min
Api Streaming	1902	9.50 h

Todos los datos extraídos con los métodos anteriormente mencionados (APIs de Twitter y Octoparse) se almacenaron en la base de datos Elasticsearch y luego hacer consultas sobre ellos. Al momento de realizar consultas se lo puede hacer por palabra o frase, en figuras 19 y 20 se observa la consulta al índice (tweets_search) donde se almacenaron los tweets para saber en cuantos documentos se encuentra la palabra Ecuador.

Figura 19 Consulta de datos almacenados utilizando la API Streaming

Figura 20 Consulta de datos almacenados utilizando la API REST

Conclusiones

Las herramientas de extracción brindan mucha utilidad al momento de realizar el proceso de recopilar datos de la web, estas aprovechan la ausencia de las llamadas APIs. Para utilizar estas herramientas depende mucho de la estructura de cada página web y su estructura de datos, en el caso de Twitter, se toma en cuenta que los datos se presentan a medida que se va avanzando en la página, debido a esto, se debe seleccionar bien el campo para que esta herramienta pueda extraer a medida que vaya avanzando la página.

Las APIs de Twitter al igual que las de otras aplicaciones web ayudan mucho al momento de realizar extracciones de comentarios, estas APIs no dependen de la estructura de su página, gracias a la gran cantidad de clases y métodos, esta actividad hace que la extracción de datos sea aparentemente sencilla.

La creación del corpus se realizó por medio de dos herramientas diferentes, las APIs de Twitter Rest y Streaming, y la herramienta web scraping Octoparse, logrando reunir un gran volumen de datos. Cabe mencionar que al momento de extraer tweets con el API Streaming llevó más tiempo que la Api Rest, en vista que depende mucho de la interactividad que tengan los usuarios en la localidad utilizada en el experimento.

Se pretende seguir extendiendo el volumen de datos en este corpus resultante, también se tiene como finalidad ceder este corpus para investigaciones futuras, donde se podrá aplicar técnicas de análisis como lo es el Procesamiento de Lenguaje Natural (PLN), Minería de opiniones (MO) entre otras.

Referencias

1. Bernhardt, J. M., Alber, J., & Gold, R. S. (2014). A Social Media Primer for Professionals: Digital Dos and Don'ts. Health Promotion Practice, 15(2), 168–172. https://doi.org/10.1177/1524839913517235
2. Díaz-Galiano, Manuel C., Martínez-Cámara, E., Ángel García-Cumbreras, M., García-Vega, M., & Villena-Román, J. (2018). The democratization of deep learning in TASS 2017. Procesamiento de Lenguaje Natural, 60, 37–44. https://doi.org/10.26342/2018-60-4
3. Díaz-Galiano, Manuel Carlos, García-Cumbreras, M., García-Vega, M., Gutiérrez, Y., Martínez-Cámara, E., Piad-Morffis, A., & Villena-Román, J. (2019). TASS 2018: The

strength of deep learning in language understanding tasks. Procesamiento de Lenguaje Natural, 62, 77–84. https://doi.org/10.26342/2019-62-9

4. Fantinuoli, C. (2016). Revisiting corpus creation and analysis tools for translation tasks. Cadernos de Tradução, 36(1), 62. https://doi.org/10.5007/2175-7968.2016v36nesp1p62
5. Fernández, J., Gutiérrez, Y., Gómez, J. M., & Martinez-Barco, P. (2015). GPLSI: Supervised Sentiment Analysis in Twitter using Skipgrams. 294–299. https://doi.org/10.3115/v1/s14-2048
6. Han, B., Cook, P., & Baldwin, T. (2014). Text-based twitter user geolocation prediction. Journal of Artificial Intelligence Research, 49, 451–500. https://doi.org/10.1613/jair.4200
7. Instituto de ingeniería del conocimiento, . Procesamiento del lenguaje natural ¿qué es? Recuperado de http://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/. (2017).
8. Martínez-Cámara, E., Martín-Valdivia, M. T., Ureña-López, L. A., & Montejo-Ráez, A. R. (2014). Sentiment analysis in Twitter. Natural Language Engineering, 20(1), 1–28. https://doi.org/10.1017/S1351324912000332
9. Mrtinez, I. M. (2015). Rojo, Palacios, Corpus de aprendices de español (CAES). Journal of Spanish Language Teaching, Oxford, v. n.2, , . https://doi.org/10.1080/23247797..1084685. 2 DOI-1, 194–200.
10. Pitkowski, E. F., & Vásquez Gamarra, J. (2009). El uso de los corpus lingüísticos como herramienta pedagógica para la enseñanza y aprendizaje de ELE. Tinkuy: Boletín de Investigación y Debate, 11, 31–51. http://dialnet.unirioja.es/servlet/articulo?codigo=3303856&info=resumen&idioma=FRE
11. Pla, F., & Hurtado, L. F. (2014). Sentiment analysis in Twitter for Spanish. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8455 LNCS, 208–213. https://doi.org/10.1007/978-3-319-07983-7_27
12. REST API. (2021). Twitter Developer Fecha de Consulta 11 de Febrero de Disponible En Httpsdevtwittercomrestpublic.

13. Schulz, A., Loza, E., Thanh, M. +, Dang, T., & Schmidt, B. (2014). Evaluating Multi-label Classification of Incident-related Tweets. CEUR Workshop Proceedings, 1141, 26–33. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/
14. Streaming, A. P. I. (2021). Twitter Developers,[fecha de Consulta 11 de Febrero de] Disponible en: https://dev.twitter.com/streaming/overview.
15. Villena-Román, J., & García-Morera, J. (2013). TASS 2013-Workshop on Sentiment Analysis at SEPLN 2013: An overview. XXIX Congreso de La Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN 2013), 50, 37–44. http://www.daedalus.es/TASS2013/papers/tass2013-overview.pdf