

Detección de Emociones y Recuperación de la Información

|  |  |
| --- | --- |
| **Docente** | Ing. Borre, Hernán Emilio |
| **Alumno** | Vallejos, Lucas Gabriel |
| **Curso** | K3551 |
| **Fecha de Entrega** | 2023-10-27 |

Análisis del Sentimiento de una Criptomoneda

En este estudio, se pretende aplicar los principios básicos de la materia realizando un análisis de sentimiento sobre una criptomoneda mediante herramientas de procesamiento del lenguaje.

Para ello, se generará un corpus aprovechando varias integraciones de motores de búsqueda para obtener páginas web que mencionen la criptomoneda seleccionada.

A continuación, se procesarán los documentos HTML para extraer la información relevante, convertirla en elementos tangibles relacionados con los sentimientos y almacenar estos datos.

Para garantizar una cobertura exhaustiva, este estudio se centrará en el inglés como lengua principal, ya que se espera que proporcione una mayor variedad de artículos sobre el tema de investigación.

Todo este proceso implica integrar los materiales del curso, las herramientas de procesamiento del lenguaje disponibles y los recursos externos para facilitar el análisis de los sentimientos.

Introducción

La tarea de entender y analizar el lenguaje es un desafío complejo debido a cómo funciona el lenguaje en sí. Una palabra puede tener diferentes significados según el contexto. Esto puede llevar a confusiones cuando tratamos de enseñar a una computadora a entender el lenguaje.

Para analizar el texto, primero se necesitó dividirlo en partes significativas **(tokens representativos)**. Esto incluye quitar signos y palabras comunes que no aportan mucho valor al análisis de los textos en cuestión, conocidas como palabras de corte.

Luego, se **clasificaron léxicamente** estos tokens en grupos según su función, por ejemplo, si son sustantivos, verbos, adjetivos, etc. Esto es importante porque una palabra puede significar cosas diferentes según su función en la oración. Una forma de hacer esto es asignar una etiqueta a cada palabra según su categoría. Para hacer esto, se utilizó una técnica llamada **Part of Speech (POS) tagging**. Esto ayuda a los algoritmos a comprender la estructura gramatical y el significado de un texto. Es mediante la probabilidad que podremos etiquetar las palabras correctamente. Aquí entra el concepto de **Naive-Bayes**. Asumimos que la probabilidad de una palabra depende solo de la palabra anterior, lo que simplifica el cálculo.

Finalmente, después de haber etiquetado todas las palabras procedemos al "**Reconocimiento de Entidades Nombradas**" (NER). Este proceso involucra identificar palabras o grupos de palabras que son nombres propios, como nombres de personas, organizaciones o fechas.

Para hacer todo esto, se usará una biblioteca de procesamiento de lenguaje natural llamada **Spacy**. Spacy analiza el texto y asigna un sentimiento basado en la categoría de las palabras, lo que nos ayuda a comprender el contenido y la opinión expresada en el texto.

Desarrollo

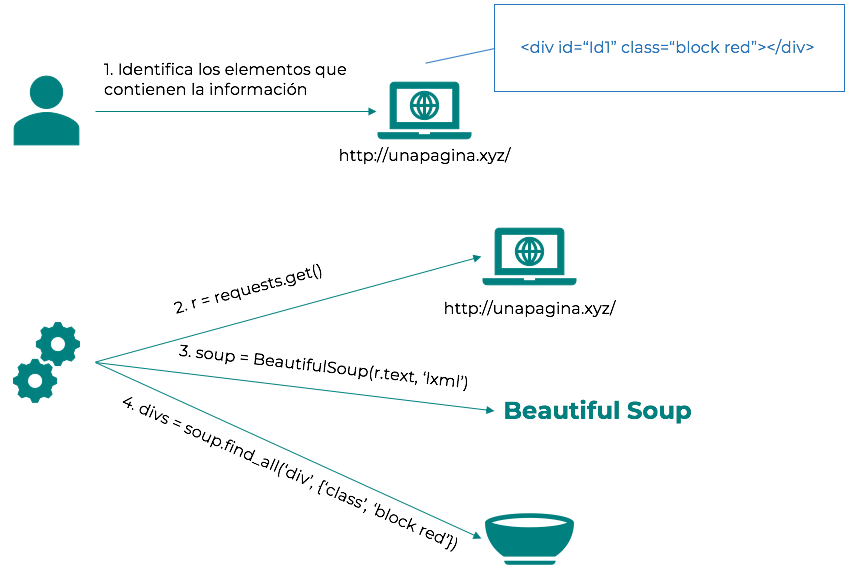
Motores de búsqueda

Para poder crear una colección de textos a analizar, se ha realizado una integración con los motores de búsqueda **DuckDuckGo**, **Yahoo** y **Bing**. Se combinaron los contenidos de estos motores de búsqueda y obtener referencias a diferentes sitios web.

Se eligió utilizar DuckDuckGo en lugar de Google para esta integración en particular debido a las diferencias en cómo manejan la privacidad de los usuarios y la recopilación de datos. DuckDuckGo se enfoca en la privacidad y no rastrea la actividad del usuario, mientras que Google recopila una gran cantidad de datos sobre las actividades de los usuarios, lo que incluye búsquedas anteriores y ubicación. Las búsquedas anteriores de Google pueden afectar el web scraping de varias maneras tales como la personalización de resultados (siendo estos no representativos cuando queramos acceder a estos) y el bloqueo de la actividad mediante su detección de bots.

Web Scraping

Estos sitios previamente combinados nos proporcionan documentos en formato HTML. Sin embargo, estos documentos pueden tener una estructura confusa y contener elementos que no son relevantes para nuestro análisis. Para solucionar esto, se utilizó la herramienta llamada **Beautiful Soup**. Esta puede detectar la estructura del documento y extraer los elementos que son útiles para nosotros.



El proceso descrito anteriormente es bastante eficaz en la mayoría de los casos, pero se vuelve más complicado debido a la diversidad de formatos que se encuentran al obtener el texto útil de los sitios web. Se trata de localizar la información relevante en la etiqueta HTML que comúnmente se llama "*Body*". Esta es donde generalmente se encuentra el contenido principal de un artículo o noticia. Por eso es importante destacar que los sitios web pueden tener estructuras muy diferentes. Algunos pueden seguir un formato estándar con una estructura clara en la etiqueta "*Body*", mientras que otros pueden ser más complejos.

Por ejemplo, algunos sitios web pueden tener una estructura muy simple donde el contenido relevante se encuentra dentro de la etiqueta "*Body*". Esto facilita el proceso de extracción. Sin embargo, otros sitios pueden utilizar diseños más elaborados con múltiples capas de etiquetas y estilos CSS. En estos casos, con nuestra herramienta Beautiful Soup lidiamos con estas variaciones para extraer la información de manera efectiva removiendo diferentes etiquetas y estilos CSS .

Es importante aclarar que no se puede garantizar que se eliminarán por completo dicho contenido innecesario. En ocasiones, algunos sitios incluso ocultan datos importantes dentro de componentes que no son fáciles de detectar. Este problema existe en varios sitios web, por lo que **nunca tendremos datos completamente limpios para analizar**.

Otro problema que enfrentamos es que algunos sitios web pueden detectar que somos un bot, y nos niegan el acceso o nos dan un documento sin valor. Cuando no obtenemos el documento que necesitamos, simplemente descartamos ese sitio web porque no tiene información para nuestro estudio. Si obtenemos un documento, pero creemos que el sitio nos detectó como un programa automatizado, sólo lo consideraremos útil si contiene **al menos una palabra clave** relacionada con nuestro tema de estudio. Para ayudarnos en este proceso, utilizaremos herramientas llamadas **NLTK** y **Scikit-learn** para convertir palabras en "tokens" (pequeñas unidades de texto). Esto es importante porque las palabras no siempre están separadas por espacios y pueden tener formas diferentes de su forma original. Por ejemplo, una palabra en plural puede significar lo mismo que la misma palabra en singular (paraguas). Si no encontramos las palabras clave que buscamos, asumimos que el sitio web acepta nuestra solicitud de acceso, pero nos envía un documento diferente para verificar que somos usuarios reales y no programas automáticos.

Análisis del Sentimiento

Luego de obtener la información de un documento, el siguiente paso es analizarla. Para hacer esto, se utilizó la herramienta llamada **Spacy** que cuenta con una inteligencia artificial especializada en procesar lenguaje natural. Sin embargo, para que funcione correctamente, necesitamos configurarla adecuadamente.

En nuestro caso, estamos utilizando un modelo de Spacy llamado "*en\_core\_web\_lg*", ya que es muy preciso para tareas como etiquetado de palabras y reconocimiento de entidades sin necesidad de herramientas adicionales. Esto significa que puede identificar las palabras y partes importantes del texto sin dificultad.

Se consideró otro modelo para este estudio y es el llamado “*en\_core\_web\_trf*”. Este modelo utiliza la arquitectura de transformers, que es una técnica avanzada en el campo del procesamiento de lenguaje natural. A pesar de ser más preciso que el modelo anterior con un diferencial porcentual de entre 1 y 2, requiere más recursos para el procesado del texto siendo un poco más lento.

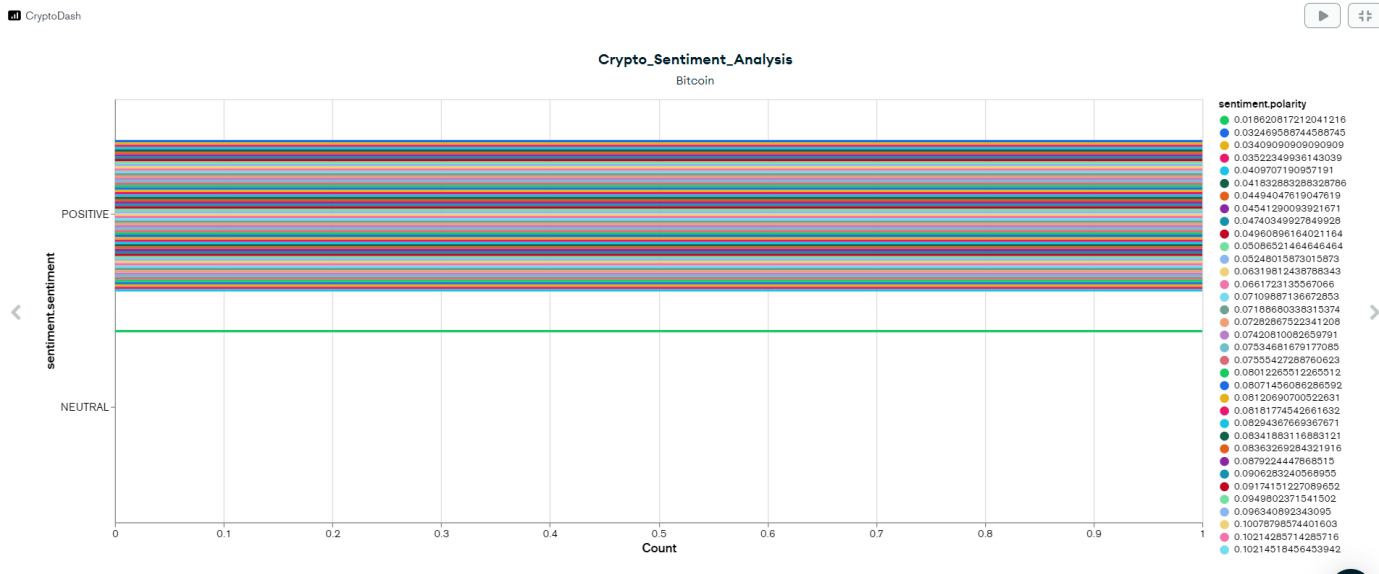
Además, usaremos una extensión de Spacy llamada "*spacytextblob*" que nos ayudará a estimar el sentimiento asociado al documento. Esta extensión se basa en el resultado proporcionado por nuestra primera configuración. En otras palabras, utilizará la información que ya obtuvimos, como las categorías de palabras y las entidades reconocidas, para evaluar las emociones presentes en el texto.

Persistencia de los resultados

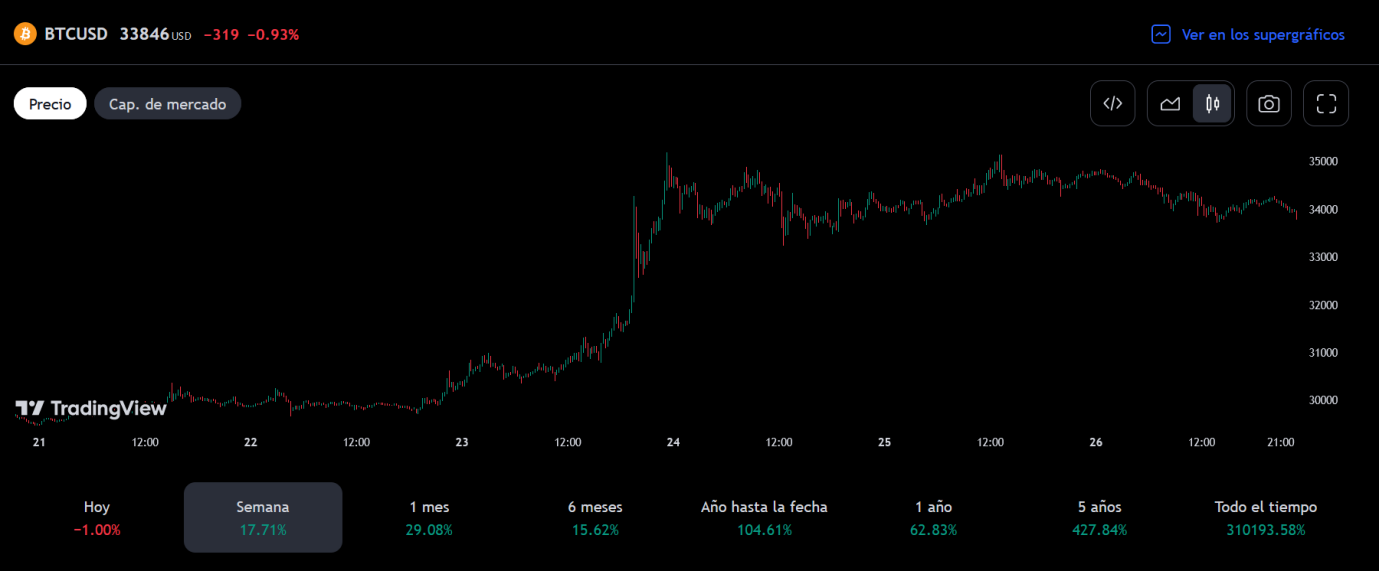
Luego de procesar los textos, el siguiente paso es adaptar y almacenar el contenido en una base de datos NoSQL. En este caso, elegimos MongoDB para esta tarea porque combina de manera efectiva la simplicidad y eficiencia en comparación con otras bases de datos no relacionales.

La información se almacenará en una base de datos llamada "crypto\_sentiment", que contendrá una sola colección con información sobre los sitios web procesados (como el título, el texto, el enlace al sitio), las entidades nombradas (NERs) detectadas sin repeticiones (junto con su tipo y la cantidad de veces que aparecen); también se almacenará el valor de polaridad del documento y el sentimiento asociado a esa polaridad, que puede ser "NEGATIVE", "NEUTRAL" o "POSITIVE".

Conclusión



Al analizar el corpora con su procesamiento asociado es posible vislumbrar una polaridad POSITIVA en su mayoría, que tiene sentido si lo comparamos con el comportamiento actual de la criptomoneda estudiada:



Sin embargo, este análisis presenta algunas perspectivas cruciales. Primero encontramos las impurezas en el texto. Se mencionó previamente la presencia de contenedores y etiquetas HTML que no aportan valor al análisis y que, de hecho, pueden introducir errores al procesar el contenido

Para mejorar esto, se sugiere ajustar el proceso de eliminación de etiquetas HTML, especialmente para manejar mejor las referencias de texto y evitar la superposición de palabras. También, se propone buscar una forma más precisa de determinar la fecha de creación del documento, lo que podría ayudar a relacionar los cambios en el valor de la criptomoneda con los sentimientos encontrados en los textos. Finalmente, para un mejor entendimiento se propone una sumarización final de las polaridades en una nueva colección de MongoDB.

Referencias bibliográficas

Laboratorios de la asignatura

* <https://github.com/LucasGVallejos/nlp_utn>

Introducción al POS Tagging

* Nombre Autor (Año publicación) Título (link)
* Atul Harsha (2022) - Understanding Part-of-Speech Tagging in NLP: Techniques and Applications (<https://www.shiksha.com/online-courses/articles/pos-tagging-in-nlp/>)
* Tiago Duque (2020) - Building a Part of Speech Tagger (<https://medium.com/analytics-vidhya/part-of-speech-tagging-what-when-why-and-how-9d250e634df6>)
* Future Mojo (2022) - NLP Demystified 4: Advanced Preprocessing (<https://www.youtube.com/watch?v=aeUE9AXO5Ss&ab_channel=FutureMojo>)

Motores de búsqueda utilizados

* Mohammad Musharraf (2019) - DuckDuckGo vs. Google: What You Need to Know (<https://medium.com/hackernoon/duckduckgo-vs-google-what-you-need-to-know-869368b08c4f>)
* search-engines v1.0.7 (<https://pypi.org/project/search-engines/>)

Introducción al Web Scraping y a Beautifulsoup

* Waleed Mousa (2023) - Mastering Web Scraping and Sentiment Analysis with Python and Machine Learning (<https://ai.plainenglish.io/mastering-web-scraping-and-sentiment-analysis-with-python-and-machine-learning-255d1d6234c5>)
* Juan José Lozano Gómez - Web scraping con Python. Extraer datos de una web. Guía de inicio de Beautiful Soup (<https://j2logo.com/python/web-scraping-con-python-guia-inicio-beautifulsoup/#scraping-beautifulsoup>)
* yuvraj\_chandra - Remove all style, scripts, and HTML tags using BeautifulSoup (<https://www.geeksforgeeks.org/remove-all-style-scripts-and-html-tags-using-beautifulsoup/>)

Librerías utilizadas para el análisis de Sentimientos

* Natural Language Toolkit (<https://www.nltk.org/>)
* scikit-learn (<https://scikit-learn.org/stable/>)
* Spacy (<https://spacy.io/> )
* spacytextblob (<https://spacy.io/universe/project/spacy-textblob> )

Material Adicional para el análisis de Sentimientos

* Spacy Models Documentation (<https://spacy.io/models/en>)
* en\_core\_web\_lg vs en\_core\_web\_trf (<https://github.com/explosion/spaCy/discussions/12849>)
* Snowball Stemmer – NLP(<https://www.geeksforgeeks.org/snowball-stemmer-nlp/>)
* Ahmad Anis, Getting Started with Sentiment Analysis using Python (<https://cnvrg.io/sentiment-analysis-python/> )
* Greg Bernhardt - Evaluate Sentiment Analysis in Bulk with spaCy and Python (<https://importsem.com/evaluate-sentiment-analysis-in-bulk-with-spacy-and-python/>)

Almacenamiento de los resultados

* How to Use Python with MongoDB (<https://www.mongodb.com/languages/python>)

Plataforma para el seguimiento de Bitcoin

* Tradingview (<https://es.tradingview.com/symbols/BTCUSD/>)