UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Buenos Aires

PROCESAMIENTO DEL LENGUAJE NATURAL

–2023–

DOCENTE: PROF. HERNAN BORRE

ALUMNO: Nicolas Goldfarb

FECHA DE ENTREGA: 27/10/2023

Consiga

En este Trabajo Práctico, como primer paso elegí las 3 APIs a utilizar para obtener información, las cuales son Reddit, Bing y Google, para comparar con los precios actuales utilizo CoinMarket. Despues decidí que la criptomoneda iba a ser Bitcoin.

Solución

Primero, empecé ingresando la identificación necesaria de Reddit, para luego realizar el scrapping necesario, agarrando los últimos post del subreddit de Bitcoin.

```
def reddit_search(query):
  news_data = []
  for submission in reddit.subreddit(query).new(limit=10):
    if submission.selftext and submission.selftext.strip():
      news_data.append({"title": submission.title, "body": submission.
  return news_data
```

```
10] results_noticias_reddit = reddit_search("bitcoin")
```

Luego escribí esta funcion de Bing, donde el parámetro es la palabra a buscar en los titulos de las noticias. Almacenando el titulo, el body, la url y el site de la noticia. Utilizo BeautifulSoup para analizar el HTML de la web

```
[14] def bing_search(query):
       headers = {
           "User-Agent":
           "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36
         }
       market = "en-US"
       regex = re.compile(r'^https://')
       html = requests.get(f"https://www.bing.com/news/search?q={query}&mkt={market}", headers=headers)
       soup = BeautifulSoup(html.text, 'lxml')

       news_data=[]
       o = {}
       for result in soup.select('.card-with-cluster'):
           article_url = result.select_one('a')['href']
           if regex.match(article_url):
             article_response = requests.get(article_url, headers=headers)
           else:
             break
           article_soup = BeautifulSoup(article_response.content, "html.parser")
           article_paragraphs = article_soup.find_all("p")
           article_title = article_soup.title.text.strip()
           article_body = " ".join(paragraph.text for paragraph in article_paragraphs)
           news_data.append({"title": article_title, "body": article_body, "url": article_url, "site": 2})
       return news_data
```

```
query = "bitcoin"
results_noticias_bing = bing_search(query)
results_noticias_bing
```

Despues guardo los textos de las noticias, asociadas con su sitio del que fue extraído.

De la misma forma que Bing con beautifulsoup, hice scrapping con Google, aprovechando su api con googlesearch:

```
def google_search(query):
  headers = {
      "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
  }
  from googlesearch import search
  help(search)
  results_urls_google = search(query,stop=2,tpe="nws")
  news_data = []
  for item in results_urls_google:
      article_url = item
      article_response = requests.get(article_url, headers=headers)
      article_soup = BeautifulSoup(article_response.content, "html.parser")
      article_paragraphs = article_soup.find_all("p")
      article_title = article_soup.title.text.strip()
      article_body = " ".join(paragraph.text for paragraph in article_paragraphs)
      news_data.append({"title": article_title, "body": article_body, "url": article_url, "site": 3})
  return news_data
```

```
results_noticias_google = google_search('bitcoin')
results_noticias_google
```

Comenzando con el análisis, Junto todas las noticias de las 3 APIs en una lista utilizando la clase Bd_news, ademas voy agregando los body de las noticias a una lista de bodys. Despues de crear un dataframe, Empiezo a analizar los sentimientos de los bodys de las noticas con SIA (Sentiment Intensity Analyzer). Calculo la polaridad del sentimiento, sea negativo, positivo o neutral, almacenadolos para cada noticia. Para almacenar sus Ner, cargué el modelo preentrenado de spaCy para el análisis del lenguaje en inglés que puede reconocer nombres, organizaciones.. .El dataframe quedaria con el título, body, sitio, url, timestamp , sentimiento y sus Ner. Aquí finalizo el análisis de las noticias por separado

Luego, primero calcule el sentimiento total del dia para almacenarlo e ir comparando con los dias anteriores.Para esto hago un dataframe con todas las palabras con los bodys juntos, despues lo spliteo para poder aplicar un filtrado utilizando expresiones regulares y eliminando palabras comunes con stopwords. Ahora ya empiezo a analizar el sentimiento de cada palabra para luego calcular el porcentaje de cada tipo de sentimientos y asi guardar el que mayor porcentaje tuvo.

Segundo, averiguo la frecuencia y el sentimiento asociado a las palabras con sentimiento positivo o negativo, asi con freqdoctor para saber la frecuencia de las palabras, solo agarrando las 20 mas comunes. Para ir terminando creo un dataframe que quedaria con la palabra, ner, timestamp, frecuencia y sentimiento.

Por ultimo, hago una solicitud a la API de CoinMarketCap para obtener información sobre el precio de Bitcoin y guardo toda la información en la Base de Datos

Base de datos

importo la biblioteca sqlite3, creando una conexión a una base de datos sqlite “pln.db”

Luego creo tres tablas "Price", "Sentimiento", y "Bitcoin_Sentimiento"

Al final inserto en la base de datos los dataframes de noticias y las palabras analizadas pedidas en el enunciado

La funcion create_price verifica si ya existe una fila en la tabla "Price" para la fecha actual. Si no existe, inserta una nueva fila con el nombre y precio de Bitcoin para la fecha actual.

La funcion create_information: inserta un nuevo registro en la tabla "Bitcoin_Sentimiento" con la fecha y hora actuales

La funcion read_price selecciona las columnas "name" y "price" de la tabla "Price" junto con la fecha

La funcion analize_price selecciona los precios de la criptomoneda entre las fechas especificadas (fecha_inicio y fecha_fin) y calcula la variación de precio. Luego, compara esta variación con el sentimiento predecido.

La variación de precio es de la moneda (del primer dia y ultimo (si es positivo o negativo))

Al final para mostrar los datos, seleccionó todas las noticas y palabras en sus propio dataframes.

Fuentes consultadas

Jorden the coder (13 de Noviembre de 2021) Learn the basics of web scraping CoinMarketCap data with Python and Beautiful Soup. https://medium.com/crypto-code/learn-the-basics-of-web-scraping-data-with-python-and-beautifulsoup-2222e6dbe117

Dmitriy Zub (Apr 25, 2021) Scrape Bing News using Python. https://python.plainenglish.io/scrape-bing-news-using-python-acc7e73b687a

Python Engineer (14 de Marzo de 2021) How to Scrape Reddit & Automatically Label Data For NLP Projects | Reddit API Tutorial. https://www.youtube.com/watch?v=8VZhog5C3bU

hurinhu GoogleNews. https://pypi.org/project/GoogleNews/