Tipología y ciclo de vida de los datos

PRA1: ¿Cómo podemos capturar los datos de la web?

Presentado por: Jhon Jairo Realpe

1. Contexto

La información se ha recolectado en el contexto de los medios de comunicación, en particular, la información/contenido/noticias generadas diariamente por el diario El Tiempo de Colombia, su dirección web es: https://www.eltiempo.com/.

2. Título

El conjunto de datos lleva por título: Actualidad y noticias en Colombia.

3. Descripción del dataset

El conjunto de datos toma como referencia 5 de los 14 ámbitos/categorías de noticias del portal web del diario el tiempo. Las categorías seleccionadas son: opinión, Colombia, política, justicia, economía y unidad investigativa. La elección de dichas categorías se hace con el objetivo de generar y analizar su contenido de cara a modelar el sentimiento/ambiente político y económico en Colombia. En cada categoría hay una noticia y cada noticia contiene un título, epígrafe y cuerpo. De este modo la estructura del conjunto de datos es:

categoría: {“title”: “título noticia”, “epigraph”: “cuerpo epígrafe”, “body”: “cuerpo de la noticia”

4. Representación gráfica

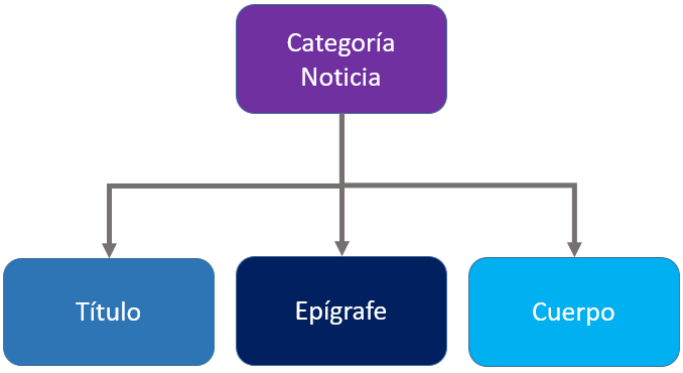

Figura 2. Representación gráfica conjunto de datos [fuente: propia]

Figura 2. Representación visual [fuente: https://www.flickr.com/photos/muledriver/14199260742/in/photostream/]

5. Contenido

El conjunto de datos está almacenado en formato json y los campos son los siguientes:

Tabla 1. Campos del conjunto de datos [fuente: propia]

Atributo	Tipo	Descripción
categoría	string	Nombre de la categoría de noticias
title	string	Título de la noticia
epigraph	string	Breve resumen de la noticia
body	string	Cuerpo de la noticia

Por otro lado, el periodo de recolección de los datos es diario.

6. Propietario

El conjunto de datos obtenido es de propiedad de la **CASA EDITORIAL EL TIEMPO S.A (CEET).**

Los pasos que se han seguido para actuar de acuerdo a los principios éticos y legales en el contexto del proyecto son:

1) Inspeccionar en el sitio web el archivo robots.txt y verificar si se permite procesos de web-scraping.
2) Determinar a partir del archivo robots.txt las restricciones que debe tener nuestra implementación al momento de realizar la búsqueda de información.

3) Lectura del aviso de la política de privacidad (https://www.eltiempo.com/politica-privacidad).
4) Lectura de los términos y condiciones (https://www.eltiempo.com/terminos-condiciones).

7. Inspiración

La extracción de datos de medios de comunicación tal como los diarios es relevante en estudios o análisis de sentimiento. Dichos estudios se pueden orientar hacia temas geopolíticos (intensión de voto de las personas y ambiente electoral), relaciones diplomáticas, el impacto que generan las noticias en los sectores productivos, el sector financiero e inversiones, también hacia el consumo y condiciones de salud de los habitantes (Hossain, Karimuzzaman, Hossain, & Rahman, 2021; Sufi, 2022).

8. Licencia

El conjunto de datos se publica bajo licencia CC BY-NC-SA 4.0 Licence, dado que:

- Es posible compartir y/o modificar los datos bajo la misma licencia.
- Se puede atribuir los datos a su legítimo propietario.
- El uso de datos no será comercial.

9. Código

Para acceder al código se debe ingresar a:

https://github.com/jhontd03/eltiemposcraper

9.1. Aspectos relevantes en el desarrollo del proyecto

El proyecto se desarrolló con el framework scrapy que está orientado a rastrear y extraer datos estructurados de sitios web (Scrapy, 2022).

El código realiza el proceso de scraping en 5 fases, cuya clase y funciones se encuentran en el archivo spiders/news.py:

1) Se definen los xpath para las categorías de noticias, enlaces, título, epígrafe y cuerpo de cada noticia.

```
HOME_URL = 'https://www.eltiempo.com/'
XPATH_MENU = '//ul[@class="default-menu"]/li/a/@href'
XPATH_LINK = '//h3/a[(@class="title page-link") or (@class="title page-link")]/@href'
XPATH_TITLE = '//div/h1[(@class="titulo") or (@class="titulo ")]/text()'
XPATH_EPIGRAPH = '//div[@class="epigraph-container lead"]/h2/text()'
XPATH_BODY = '//div[@class="articulo-contenido"]/div/p[(@class="contenido ") or (@class="contenido")]/text()'
```

2) Se configuran los parámetros generales del programa tal como:

- User-agent.
- Número máximo de peticiones.
- Retardo entre cada petición.
- Profundidad en la búsqueda de enlaces.
- Tiempo de espera en la respuesta del servidor.
- Tipo de formato a exportar.

```
custom_settings = {
    # se configura el formato a exportar
    'FEEDS': {
        f'dataset/news_eltiempo_{date}.json': {
            'format': 'json',
            'overwrite': True,
            'encoding': 'utf-8'
        }
    },
    # Se configura el máximo numero de peticiones concurrentes
    'CONCURRENT_REQUESTS': 1000,
    # Se configura el uso máximo de memoria
    'MEMUSAGE_LIMIT_MB': 2048,
    'MEMUSAGE_NOTIFY_EMAIL': ['jrealpe0@uoc.edu'],
    # Se respetan las reglas del archivo robots.txt
    'ROBOTSTXT_OBEY': True,
    # Se configura el user-agent
    'USER_AGENT': 'Mozilla/5.0 (iPad; CPU OS 12_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Mobile/15E148',
    # Se configura el tiempo de espera ante petición al servidor web
    'CLOSESPIDER_TIMEOUT': 10,
    # Se configura la máxima profundid en links a explorar
    'DEPTH_LIMIT': 5,
    # Se genera un retardo de 500ms para cada petición
    'DOWNLOAD_DELAY': 0.5,
}
```

3) Se crea la función parse la cual realiza los siguientes pasos:

- Obtiene los links de las categorías de noticias.
- Seleccionar solo 5 categorías de interés.
- Se hace el request a cada link y se hace el llamado a la función parse_page.

```python
def parse(self, response):
    # Crea directorio donde se almacenará el dataset
    if not os.path.isdir('dataset'):
        os.mkdir('dataset')

    # Se obtiene lista con categoria de noticias
    links_pages = response.xpath(XPATH_MENU).getall()

    sel_pages= ['/opinion', '/economia', '/colombia',
                '/justicia', '/politica', '/unidad-investigativa']

    # Se seleccionan solo las categorias de interés
    links_pages = [x for x in links_pages if x in sel_pages]

    # Se hace un request a cada link de categoría de noticias
    # y se hace llamado a función parge_page
    for links in links_pages:
        url = response.urljoin(links)
        yield scrapy.Request(url, callback=self.parse_page)
```

4) Obtención de los enlaces de cada noticia y llamado de la función parse_link.

```python
def parse_page(self, response):
    # Se crea una lista con los links de cada noticia
    links_articles = response.xpath(XPATH_LINK).getall()
    # Se hace requesta a cada link y se hace llamado a la función parse_link
    for url in links_articles:
        yield response.follow(url, callback=self.parse_link, cb_kwargs={'url': response.urljoin(url)})
```

5) Obtención de la categoría, título, epígrafe y cuerpo de la noticia y volcado de información a fichero json.

```python
def parse_link(self, response, **kwargs):
    # Se obtiene la información correspondiente para cada noticia
    url = kwargs['url']
    category = url.split("/")[3]

    sel_category= ['opinion', 'economia', 'colombia',
                   'justicia', 'politica', 'unidad-investigativa']

    # Se filtran categorias de interés
    if category in sel_category:
        title = response.xpath(XPATH_TITLE).get()
        epigraph = response.xpath(XPATH_EPIGRAPH).get()
        # se obtiene el cuerpo de la noticia
        body = response.xpath(XPATH_BODY).getall()
        # Dado que el cuerpo de la noticia tiene múltiples parrafos
        # se concatenan.
        body = ' '.join(body)
        # Se volca la información a fichero json
        yield {
            category: {
                'title': title,
                'epigraph': epigraph,
                'body': body
            }
        }
```

Por otro lado, en el desarrollo del proyecto no se encontraron dificultades, salvo el aprendizaje de xpath y uso del framework scrapy, este último, contiene muchas funcionalidades y puede ser confuso el entendimiento y alcance de cada uno de los elementos que constituyen un proyecto generado por este framework. Sin lugar a dudas, se requiere de más tiempo para profundizar y explotar sus capacidades.

10. Contribuciones

Contribuciones	Firma
Investigación previa	Jhon Jairo Realpe
Redacción de las respuestas	Jhon Jairo Realpe
Desarrollo del código	Jhon Jairo Realpe
Participación en el vídeo	Jhon Jairo Realpe

11. Enlace Video

El video se puede visualizar en:

https://drive.google.com/file/d/1tbYilWP_CCJRvlu3NugoOXj6nqCfXIM9/view?usp=share_link

12. Enlace repositorio Zenodo

Es repositorio del conjunto de datos se encuentra en: https://doi.org/10.5281/zenodo.7317125

Bibliografía

Hossain, A., Karimuzzaman, M., Hossain, M. M., & Rahman, A. (2021). Text Mining and Sentiment Analysis of Newspaper Headlines. *Information*, *12*(10), 414. https://doi.org/10.3390/info12100414

Scrapy. (2022). Scrapy 2.7 documentation¶. Retrieved November 12, 2022, from https://docs.scrapy.org/en/latest/

Sufi, F. K. (2022). Identifying the drivers of negative news with sentiment, entity and regression analysis. *International Journal of Information Management Data Insights*, *2*(1), 100074. https://doi.org/10.1016/j.jjimei.2022.100074