Web Scraping de imágenes y videos

con phyton

Henri Gerard Gabert Hidalgo

Universidad de Costa Rica

Proyecto Personal de:

Herramientas de Ciencias de datos II

Profesor:

Luis Alberto Potoy Juárez

Fecha:

31 de mayo del 2024

El web scraping es un proceso de extracción de contenidos y datos de sitios web mediante software según kinsta, al ser mediante software esta extracción puede ser realizada en diversos lenguajes de programación, para extraer cualquier tipo de datos, desde bases de datos, como imágenes o audiovisuales en general.

Existen muchas razones por las cuales preferiríamos el que los datos fueran extraídos por un software, dado que eso nos da la posibilidad de automatizar el proceso, de tal forma que se realice más eficientemente y sin necesidad de estar una persona detrás de toda esa obtención de datos.

El web scraping es ampliamente utilizado en páginas web que realizan comparación de hoteles, vuelos, las que te permiten rentar automóviles en distintas compañías, también las páginas que te permiten descargar videos de YouTube, incluso la creación de inteligencias artificiales tiene dentro del entrenamiento de estas, la realización de web scraping.

Un caso hipotético donde se podría utilizar web scraping:

Quieres realizar un negocio revendedor, pero dado que tu presupuesto es poco para comprar al mayoreo y vender al detalle, para realizar las compras de tus productos lo realizaras por internet localizando aquellos lugares donde el precio sea más asequible, este proceso lo puede realizar una persona entrando a diversas páginas web y comparando sus precios, posteriormente realizando la compra y reventa, mediante la vía que considere conveniente; sin embargo este proceso de ingresar a diversas páginas web, descargado de datos y comparación también puede ser realizados de forma automática al programarlo y es aquí donde entra en juego el web scraping, puesto que el ingresar a páginas web y descargar los datos de estas páginas es el motivo principal de existencia de este proceso.

Como se mencionó el web scraping no solamente se utiliza para descargar bases de datos, si no también imágenes, videos y audios, por lo que otro caso donde se puede usar es por aquellas personas que se dedican a la creación de contenido, como los Youtuber, para realizar un sistema que descargue los memes más actuales de reddit o una página web en la que surja contenido de provecho para utilizar en la edición de sus videos.

Es importante prestar atención a que existen leyes que defienden los derechos de autor, derechos a la imagen personal, por lo que ya sea en datos, imágenes o audiovisuales en general, se debe tener cuidado de no infringir estos, además no todas las páginas web tienen permitido realizar este proceso automatizado, por lo que tienen ciertos bloqueos para los softwares que realizan el proceso de web scraping, aunque existen formas programáticas de evitar estos bloqueos para poder descargar la información que necesitamos.

Existen distintas librerías para realizar web scraping, sin embargo, yo me centré en la utilización de BeautifulSoup, la cual nos permite crear un objeto tipo beautifulSoup y poder utilizar todos los métodos de la clase.

Lo primero que realizo en el método “download\_images” es el definir una URL de las que voy a realizar las consultas, esta construcción la simplifico, pidiéndole al usuaria que ingrese una única palabra clave, sin embargo si quisiera hacerla más compleja, solamente tengo que pasar el texto escrito en la consola a formato HTTP (Hypertext Transfer Protocol), esto por decirlo de alguna forma es el lenguaje de comunicación utilizado para la transferencia de información en el internet, además se utiliza el URL con el buscador de Bing dado que existe una gran compatibilidad entre Microsoft y el buscador, posteriormente se define un “header” que el cuál simula el estar realizando la consulte desde un buscador en este caso Firefox pues como dije anteriormente, muchas páginas tienen un bloqueo para impedir la automatización de estos procesos y esta es una solución para evitar estos problemas, se formula la variable “response” al utilizar un método perteneciente a la clase request , pasándole el url en el cual vamos a hacer la consulta y el “header” para realizar la consulta fingiendo que es un buscador.

Luego se inicializa la variable “soup” como un objeto de la clase Beautifulsoup para poder analizar el contenido HTML almacenado en la variable response, con el método “.get\_text()” extrae texto, pero para mí código se utilizó un método que nos permite buscar objetos más específicos, “.find\_all("img")” con el cual busca todas las etiquetas <img> .

En HTML existen muchos tipos de etiquetas para estructurar y dar formato al contenido de una página web, las etiquetas de estructura:

<html>: Define el comienzo y el final del documento HTML.

<body>: Contiene el contenido principal de la página web, como texto, imágenes y otros elementos.

<div>: Se utiliza para crear divisiones o secciones en una página web.

Etiquetas de texto: define párrafos, definen encabezados de diferentes niveles prioridades (Títulos, subtítulos…), aplican estilos…

Etiquetas de lista: lista desordenada, lista ordenada…

Y las etiquetas que nos conciernen son:

Etiquetas de enlace y multimedia:

<img>: Define una imagen en la página web.

<video>: Define un elemento de video en la página web.

<audio>: Define un elemento de audio en la página web.

Por otro lado, el <mimg> es una clase de CSS, estas clases se utilizan para aplicar estilos a elementos HTML, por lo que con “. find\_all('img', {'class': 'mimg'})” se buscan las imágenes que tengan el estilo ‘mimg’.

También en esta función entra en juego el paquete “os” el cual nos permite interactuar con el sistema operativo, dado a que está diseñado para ser multiplataforma esta librería de phyton funciona en Windows, macOS y Linux, en mi código específicamente verifica si existe la carpeta donde se van a guardar las imágenes o videos y en caso de no ser así la crea.

El ciclo “while” dentro del método “download\_images” nos permite iterar sobre las imágenes encontradas en el HTML y realiza el número de iteraciones necesarias para descargar las imágenes solicitadas, verifica con el “if img\_url” si se puede obtener la URL y no está vacía o no está en un formato incompatible y en caso de que todo esté correcto y se tenga en la variable “img\_url” la información deseada procede a ir guardándolas, en caso de ocurrir un problema manda un mensaje de error, en esta parte del código había realizado un “for” pero en caso de que las primereas imágenes estuvieran en un formato incompatible solo descargaba, el número de imágenes compatibles en las primeras “num\_images” iteraciones (variable de cantidad de imágenes que desea el usuario).

En HTML el atributo “src” (fuente) se utiliza en la etiqueta <img> para especificar la URL , con “os.path.join()” construye el archivo con un nombre que nos ayude a identificar las imágenes que descargamos, el +1 es para evitar duplicados, en “img\_data” es la variable en la que se guardan los datos binarios del URL que contenía a la imagen y se utiliza “open” de los paquetes estándar de phyton para escribir la imagen a 'w' indica que el archivo se abrirá en modo de escritura, y la 'b' indica que será en modo binario, este comando es útil especialmente cuando se trabajará con archivos que no son de texto, como imágenes o archivos de audio.

Para la importación de video utilicé el paquete “pytube”, en mi método “download\_video” lo primero es crear un objeto tipo YouTube, Filtra y selecciona la primera corriente (stream) disponible en formato mp4 (esto debería elegir la mejor calidad disponible en mp4), descarga el video en la ubicación especificada y le pone nombre al archivo y por último imprime un mensaje de éxito.

Mi método “On\_search” básicamente es el que interpreta lo que el usuario escribió en consola hay un par de “if ” por si acaso el usuario escribe de forma equivocada números negativos o 0 y por si acaso no escribe nada, también un “if y elif” para que la persona escoja si lo que desea descargar es una imagen o video, según se escoja se llama a los métodos anteriormente descritos (“download\_images” y “download\_video”).

Lo último a describir sería la interfaz con el usuario, la cual fue construida utilizando el paquete “tkinter”, con ella se pueden realizar cuadros de texto, cajas de mensajes para informar al usuario de posibles errores o el éxito de las operaciones, se le pueden poner imágenes de fondo para embellecer la interfaz, junto con cambiar la fuente, tamaños y color de las letras, colocar botones para poder ejecutar código y otras clases de widgets como el utilizado para seleccionar entre video e imagen que se le conoce como “radiobutton” , por lo que esas fueron las cosas que realicé para le interfaz.

Con lo que respecta a los resultados son satisfactorios, pues el objetivo de mi proyecto individual era la realización de una interfaz, que pudiera realizar Web-Scraping de imágenes y videos, utilizando 5 librerías distintas pude lograr ese objetivo, con mi persona aprendiendo en el proceso y espero que, en la presentación, mis compañeros también.

El código se puede mejorar permitiendo al usuario escribir más de una palabra clave, eso convirtiendo se realizaría convirtiendo la cadena que escriba en código HTTP y realizando lo demás de la misma manera, otra forma de mejora es implementando la descarga de Gifs o realizando una mejor interfaz en la que al usuario le muestre el resultado del HTML resultante y que él escoja las imágenes que desea descargar.

Para información teórica sobre web scraping:

[¿Qué Es el Web Scraping? Cómo Extraer Legalmente el Contenido de la Web (kinsta.com)](https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/)

Para la interfaz con tkinter:

<https://adictosaltrabajo.com/2020/06/30/interfaces-graficas-en-python-con-tkinter/>

<https://docs.python.org/es/3.10/library/tk.html#index-0>

Importar con BeautifulSoup:

<https://realpython.com/python-web-scraping-practical-introduction/#install-beautiful-soup>

Paquete Os:

<https://docs.python.org/es/3.10/library/os.html>

Paquete pytube:

<https://pypi.org/project/pytube/>