Extrakcia informácii z webu pre analýzu dát pomocou Python*

Štefan Kučerák
Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xkucerak@stuba.sk

5. október 2023

Abstrakt

Extrakcia informácii z webu v angličtine označovaná ako "Web scraping"je automatický spôsob získavania údajov zo stránky. Umožňuje získať presné informácie v krátkom čase. Získané dáta sú pripravené na ďalšie spracovane.

1 Úvod

Webstránky sú prevažne určené pre ľudí a tomu zodpovedá aj ich grafický dizajn. Ak chceme použiť hodnotné informácie z webstránky pre našu aplikáciu nastáva problém z ich extrakciou. Manuálna extrakciou týchto informácii by bola časovo náročná a s veľkou pravdepodobnosťou by nastala chyba pri kopírovaní. Riešením tohoto problému je automatizovaný zber informácii, ktorý priblížim v nasledujúcom článku. Celý proces extrakcie dát sa dá vyhotoviť pomocou rôznych programovacích jazykov. Pre tento článok som vybral programovacom jazyku Python, pretože je v dnešnej dobe dosť populárny a využíva sa v oblasti analýzy dát.

Vysvetlenie princípu extrakcie dát z webu je v časti 2. V časti 3 sú príklady využitia tejto metódy v reálnom svete. Knižnice pre extrakciu dát sú v časti 4 a príklad implementácie pomocou jednej z knižníc je v časti 5.

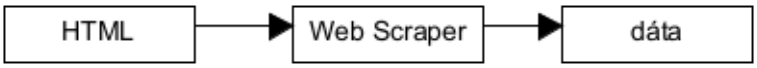
(poznámka: Článok je stále v príbehu písania, zatiaľ sú skôr zachytené myšlienky, ktoré by som chcel použiť. Vety ešte nie sú úplne správne sformulované. Môj východzí článok je: [1])

2 Web Scraping

Cieľom je získať dáta a následne ich analyzovať, napríklad pre výhodu na trhu. Získavanie dát je zložitý proces obsahujúci viacej krokov začínajúc výberom aké dáta vlastne chceme zbierať, ich usporiadaním, odstraňovaním nesprávnych

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2023/24, vedenie: Ing. Mohammad Yusuf Momand, MSc.

dát, opätovnou analýzou použitím algoritmov [8]. Získavanie veľkého množstva dát pomocou ľudskej sily by bolo veľmi obtiahne a vzniká priestor na chybu [1]. Tento problém rieši Web Scraping a to tak že pomocou automatizácie je možne získavať tieto dáta rýchlejšie a univerzálnejšie. Web scraping je používaný spoločnosťami ale aj jednotlivcami, ktorý chcú využiť voľne dostupné informácie [1]. Príklady jeho využitia sú napríklad: sledovanie ceny výrobku, návštevnosti stránky, čítanie komentárov na stránke, zistenie hypertextových odkazov na danej stránke a mnoho ďalších [4].



Obr. 1: Princíp extrakcie

Web scraper v tomto prípade knižnica pre Python stojí medzi html stránkou a dátami o ktoré máme záujem (Obr. 1). Na začiatku musíme mať nejakú cieľovú stránku z ktorej chceme dáta získať. Scrapery vyhľadávajú v html podla html tagov ako napríklad: h1, div, span [7].

3 Príklady reálneho využitia tejto metódy

Jedným z príkladov je článok "Phishing Web Page Detection using web Scraping“ [3], v ktorom sa zaoberajú automatizovaným odhaľovaním phishing stránok. Ich riešenie na odhaľovanie takýchto stánok má presnosť detekcie viac ako 98%. Pri tomto projekte bol požitý Python a knižnica Beautiful Soup(časť 4).

Ďalším z příkladov je článok "Food Genie, Recipe Search Algorithm using Web Scraping" [6], ktorý využiva web scraping na porovnanie indegriencíi, ktoré majú ísť do rovnakého receptu. Rieši problém toho že naprieč rôznymi stránkami má recept iný postup alebo sú použité iné suroviny. Napríklad pri tomto projekte bol využitý Python a knižnica scrapy(časť 4).

Ako posledný príklad by som uviedol článok "NEWSONE- AN AGGREGATION SYSTEM FOR NEWS USING WEB SCRAPING METHOD" [5], ktorý požíva web scraping na extrakciu správ z rôznych web stránok v rôznych jazykoch a sústreďuje ich na platformu "NewsOne". Pri tomto projekte sa taktiež využíva Python a knižnica Beautiful Soup(časť 4).

4 Knižnice pre získavanie dát

Zoznam knižníc, ktoré môžu byť použité na extrakciu dát z webu pre programovací jazyk Python:

- Requests[1] je jednoduchá knižnica ktorá umožňuje vytvoriť http požiadavky a to post aj get. Táto knižnica je najčastejšie používaná na získanie html zo stánky. Ak by sme chceli použiť len samotnú knižnicu museli by sme si na vyhľadávanie napísať vlastný algoritmus.

[1] `https://pypi.org/project/requests/`

- Beautiful Soup[2] táto knižnica obsahuje nástroje na vyhľadávanie, upravovanie a iteráciu v html alebo xml. Podmienkou pre fungovanie je vstup vo formáte html alebo xml a pre ten sa najčastejšie využíva knižnica Requests.

- Scrapy[3] je rýchly framework pre prehľadávanie webu na vysokej úrovni, ktorý sa používa na prehľadávanie webových stránok a extrahovanie štruktúrovaných údajov. Dá sa použiť na širokú škálu účelov, od získavania dát až po monitorovanie a automatizované testovanie. Výhodou oproti Beautiful Soup je že knižnica obsahuje nástroj na extrakciou html zo stránky.

- Selenium[4] je knižnica, ktorá umožňuje priamo pomocou prehliadača interagovať so stránkami. Podporované prehliadače Firefox, Chrome, Internet Explorer.

- MechanicalSoup[5] Knižnica na automatizáciu interakcie s webovými stránkami. MechanicalSoup automaticky ukladá a odosiela súbory cookie, sleduje presmerovania a môže sledovať odkazy a odosielať formuláre. Avšak nepodporuje JavaScript.

5 Jednoduchý príklad požitím knižnice Beautiful Soup

```
1  import requests
2  from bs4 import BeautifulSoup
3
4  page = requests.get("https://www.google.com/")
5  soup = BeautifulSoup(page.content, "html.parser")
6
7  print(soup.title.string)
8
```

Obr. 2: Program

Na Obr. 2 je príklad kódu, ktorý zo stránky `https://www.google.com/` extrahuje názov stánky a teda tag title. Ako prvé sa spraví požiadavka pomocou knižnice Requests, ktorá vráti web stránku vo formáte html. Na html sa použije knižnica Beautiful Soup, ktorá vie pracovať s jednotlivými tagmi v html súbore. Knižnici dáme požiadavkou na vypísanie názvu stránky v reťazci. Následne dostaneme výstup "Google". Tento princíp je uvedený v Obr. 1. Podobný postup sa dá využiť aj napríklad na sledovanie ceny produktu a to tak že nájdeme html tag, ktorý obsahuje symbol meny a teda aj číselnú hodnotu.

[2] `https://pypi.org/project/beautifulsoup4/`
[3] `https://pypi.org/project/Scrapy/`
[4] `https://pypi.org/project/selenium/`
[5] `https://pypi.org/project/MechanicalSoup/`

6 Záver

Výsledkom mojej práce je článok ktorý približuje tému extrakcie informácii z webu pomocou programovacieho jazyka Python. A taktiež projekt na predmet Metódy inžinierskej prace.

Spoločenské súvislosti.

Historické súvislosti.

Technológia a ľudia.

Udržateľnosť a etika. Tento spôsob získavania informácii môže byt neetický, pretože stránka môže predávať svoje dáta za peniaze a tak obchádzame tuto peňažnú bariéru. Týmto problémom sa zaoberajú mnohý ľudia a jedným z príkladov je článok [2]., ktorý navrhuje rôzne spôsoby ako zabrániť extrakcii dát.

Literatúra

[1] Ayat Abodayeh, Reem Hejazi, Ward Najjar, Leena Shihadeh, and Rabia Latif. Web scraping for data analytics: A beautifulsoup implementation. In *2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, pages 65–69, 2023.

[2] Chandan Biswas, Rahul Mallick, Subrata Paul, and Dipta Mukherjee. Solution to web scraping. In *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, pages 1–5, 2023.

[3] Mallika Boyapati and Ramazan Aygun. Phishing web page detection using web scraping. In *SoutheastCon 2023*, pages 167–174, 2023.

[4] Rahul Chauhan, Ayush Negi, and Mahesh Manchanda. An extensive review on web scraping technique using python. In *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pages 1134–1138, 2023.

[5] K. Sundaramoorthy, R. Durga, and S. Nagadarshini. Newsone — an aggregation system for news using web scraping method. In *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 136–140, 2017.

[6] Nikhil Suwalka, Nishit Shanbhag, Sakina Salmani, and Pooja Raundale. Food genie, recipe search algorithm using web scraping. In *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–6, 2023.

[7] S Thivaharan., G Srivatsun., and S Sarathambekai. A survey on python libraries used for social media content scraping. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pages 361–366, 2020.

[8] David Mathew Thomas and Sandeep Mathur. Data analysis by web scraping using python. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 450–454, 2019.