Pannon Egyetem   
Műszaki Informatikai Kar  
Projekt Alapú Szoftverfejlesztés (VEMISAM233P/2020/21/2)  
Programtervező informatikus MSc

**Publikációs listák szinkronizálása**

2021

(RSZT 3. téma)

Témavezető: Szekér Szabolcs

Csapatattagok:

Szászfai Júlia (GO45XL)  
 Csizmazia Máté (XI32IS)  
 Pataki Miklós (BQPM6L)

**Tartalom**

[**1.** **MTMT és Google Scholar rendszerek ismertetése** 3](#_Toc66132465)

[**1.1.** **MTMT rendszer** 3](#_Toc66132466)

[**1.2.** **Google Scholar rendszer** 3](#_Toc66132467)

[**2.** **Felhasználóhoz tartozó publikációs lista lehívása MTMT és Google Scholar rendszerekből** 3](#_Toc66132468)

[**3.** **Google Scholar API vizsgálata, kiértékelése (MTMT nem biztosít API-t)** 3](#_Toc66132469)

[**3.1.** **Scraper API** 3](#_Toc66132470)

[**3.2.** **SERP API** 3](#_Toc66132471)

[**3.3.** **SerpWow** 3](#_Toc66132472)

[**3.4.** **Scale SERP** 3](#_Toc66132473)

[**3.5.** **Publish or Perish** 3](#_Toc66132474)

[**4.** **Web scraping technológia bemutatása** 4](#_Toc66132475)

[**4.1 Web scraping fogalma** 4](#_Toc66132476)

[**4.2 Web scraping története** 5](#_Toc66132477)

[**4.3 Technikák** 5](#_Toc66132478)

[**4.3.1 Emberi másolás-beillesztés** 5](#_Toc66132479)

[**4.3.2 Mintaszöveg egyezés** 5](#_Toc66132480)

[**4.3.3 HTTP programozás** 6](#_Toc66132481)

[**4.3.4 HTML elemzés (parsing)** 6](#_Toc66132482)

[**4.3.4 DOM elemzés** 6](#_Toc66132483)

[**4.3.5 Függőleges felhalmozás** 6](#_Toc66132484)

[**4.3.6 Szemantikai annotáció felismerése** 6](#_Toc66132485)

[**4.3.7 Számítógépes látás – weboldal analízis** 7](#_Toc66132486)

[**4.4 Selenium** 7](#_Toc66132487)

[**5.** **Azonos feltételű kereső megvalósítása MTMT és Google Scholar rendszerekben** 7](#_Toc66132488)

[**6.** **Adatok kigyűjtése, találatok megjelenítése** 7](#_Toc66132489)

[**7.** **Találatok összehasonlítása és különbségek keresése** 7](#_Toc66132490)

[**8.** **Automatikus/szoftverrel történő összehasonlítás előnye a manuálissal szemben** 7](#_Toc66132491)

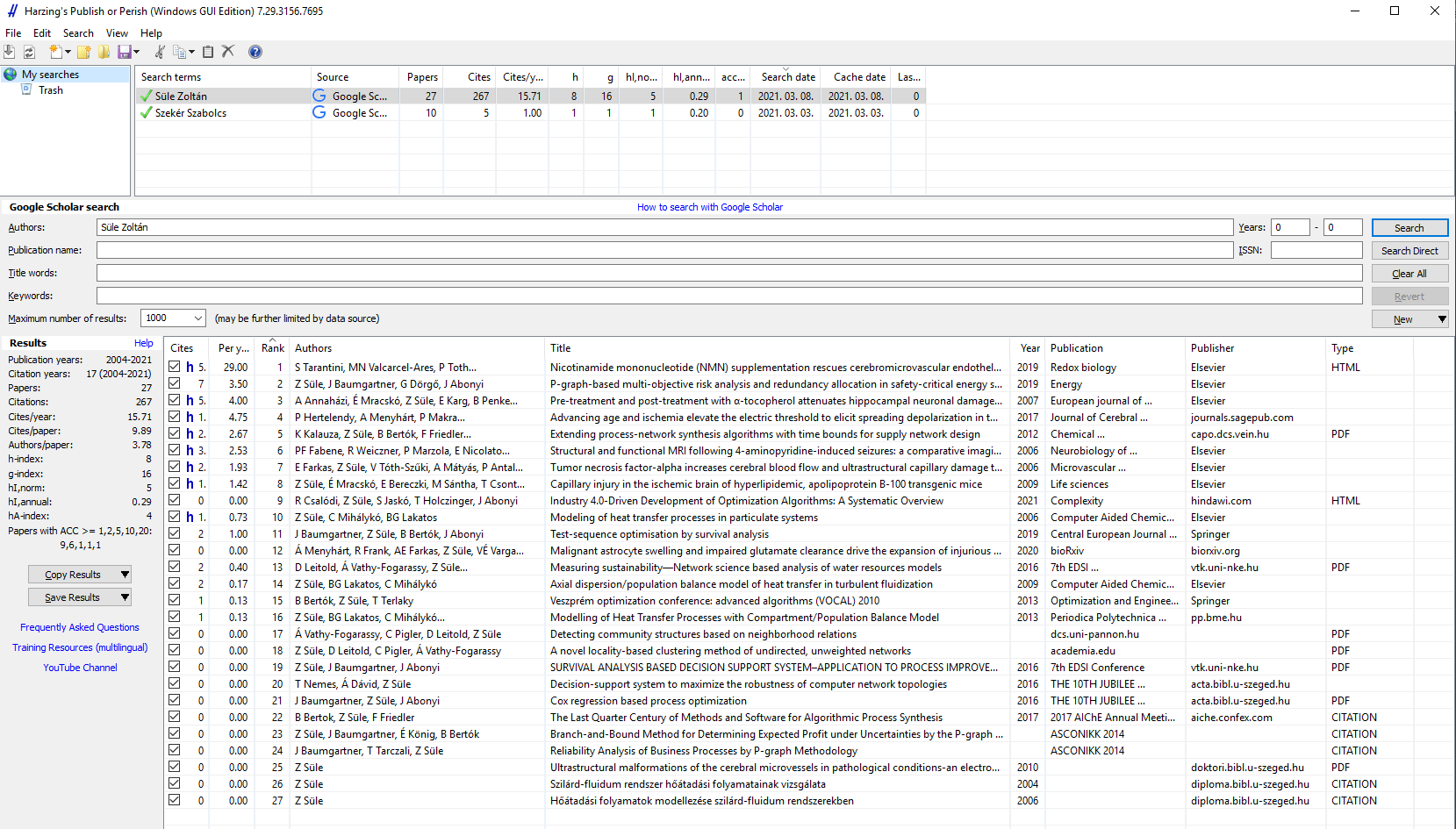
[**9.** **Adatok összehasonlítása és kiértékelése, különbségek felfedezése** 7](#_Toc66132492)

[**10.** **Irodalomjegyzék** 8](#_Toc66132493)

1. **MTMT és Google Scholar rendszerek ismertetése**
   1. **MTMT rendszer**
   2. **Google Scholar rendszer**
2. **Felhasználóhoz tartozó publikációs lista lehívása MTMT és Google Scholar rendszerekből**
3. **Google Scholar API vizsgálata, kiértékelése (MTMT nem biztosít API-t)**

<https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/>

* 1. [**Scraper API**](https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/#scraperapi)
  2. [**SERP API**](https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/#serpapi)
  3. [**SerpWow**](https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/#serpwow)
  4. [**Scale SERP**](https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/#scaleserp)
  5. [**Publish or Perish**](https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/#publishorperish)



1. **Web scraping technológia bemutatása**

**4.1 Web scraping fogalma**

Az angol szótár szerint a scraping jelentése kaparás. A szó szerinti jelentés nem is áll nagyon messze a technika lényegétől. A web scraping, más néven web harvesting vagy web data extraction egy automatizált adatbányászati vagy adatgyűjtési módszer, ami az adat scraping technikát használja weblap forráson. Az adat scraping olyan technika, amelyben egy számítógépes program egy másik program, ember által olvasható kimenetéből, kigyűjti az adatokat. Ez egy formája a másolásnak, amelyben meghatározott adatokat gyűjtünk ill. másolunk a webről későbbi elemzés céljából. Ezeket az adatokat akár egy adatbázisba vagy táblázatba is letárolhatjuk. A web scraping szoftver közvetlenül elérheti a webet a HTTP-t (Hypertext Transfer Protocol) vagy egy web böngészőt használva.  
A szemantikus web, egy kiterjesztése a World Wide Webnek a World Wide Web Consortium (W3C) által lefektetett szabványokon keresztül. A célja, hogy az Interneten fellelhető adatok gép által olvasható legyen.  
Az adatok webje terminológiát, maga Tim Berners-Lee (W3C elnöke) vezette be, aki 1999-ben úgy nyilatkozott, hogy „Van egy álmom a webről, ahol a számítógépek alkalmasak elemezni a weben levő összes adatot, tartalmat, linkeket és tranzakciókat emberek és számítógépek között. A szemantikus web, ami ezt lehetővé teszi, de amikor ez megtörténik, akkor a napi működése a kereskedéseknek, bürokráciának vagy a mindennapi életűnket egymás között beszélő gépek fogják kezelni.”  
Tim Berners-Lee álma még csak részben teljesült, de a weben levő tartalmak elemzésében a web scraping technológia nagy segítséget nyújt.  
Meg kell említenünk, hogy léteznek web scrapinget megakadályozó, különféle technikák, amik valóban megnehezítek a web oldalakról gyűjtött információkat, azonban ezeknek jogos céljuk is lehet; például a különféle adatvédelmi aggályok.

**4.2 Web scraping története**

A World Wide Web 1989-es megszületése után az első web „robot”, a World Wide Web Wanderer 1993 júniusában jelent meg. Ezt a Perl programot az MIT-n fejlesztett Matthew Gray, melynek célja a World Wide Web méretét megmérni. Indulásakor csupán 130 db weboldalt tartalmazott. Wanderer megjelenése után több ehhez hasonló program jelent meg.   
JumpStation például ugyanezen év végén jelent meg, ami egy „crawler”-alapú kereső motor volt. A crawler (másnéven: spider vagy spiderbot) szisztematikusan böngészi a webet tipikusan indexelés céljából. Ez egy újabb nagy lépés volt a web fejlődésének történetében, mert ez lett az alapja a weben ma is használatos keresőknek. Napjainkban is futnak ilyen crawlerer-alapú keresőmotorok; a legismertebb pl.: Google vagy a Bing.  
2000-ben megjelent az első web API (Application Programming Interface). Az API által szolgáltatott építőelemek segítségével már jelentősen könnyebb volt programokat fejleszteni. Ekkor a Salesforce és az eBay indította el a saját API-át, amivel biztosította a nagyközönség számára az adataikhoz a hozzáférést vagy akár letöltést. Ezután már több weboldal is biztosított API-t a nyilvános adatbázisukhoz való hozzáféréshez.

**4.3 Technikák**

**4.3.1 Emberi másolás-beillesztés**

A legegyszerűbb módja a web scrapingnek a weboldalon található adatok manuális másolása és beillesztése például egy szövegfájlba vagy egy táblázatba. Időnként még a legjobb web scraping technológiák sem helyettesíthetik az emberi, manuális vizsgálatot, másol-beillesztést és időnként csak ez az egy működő megoldás, hogy a kifejezetten a gép általi automata web scraping-et megakadályozó weboldalakról adatokat gyűjtsünk.

**4.3.2 Mintaszöveg egyezés**

Egy egyszerű, de hatékony megoldás lehet az információk kinyerésére a weboldalakról, ami a Unix grep parancsán vagy reguláris kifejezés egyezést használó nyelvek használatával (pl.: Perl vagy Python).

**4.3.3 HTTP programozás**

Statikus vagy dinamikus weboldalak lekérdezhetőek HTTP kérések post-olásával a távoli webszervernek socket programozással.

**4.3.4 HTML elemzés (parsing)**

Sok weboldalnak hatalmas gyűjteménye van dinamikusan létrehozott oldalakból, amiknek struktúrált forrása van, hasonlóan az adatbázisokhoz. Az azonos kategóriájú adatokat egy szkript vagy sablon tipikusan hasonló oldalakra kódolta. Az adatbányászatkor a program felfedezi ezeket a sablonokat abban az információ forrásban és kigyűjti a tartalmát és lefordítja egy relációs formára, ezt wrapper-nek hívjuk. A wrapper algoritmus feltételezi, hogy a input oldalak egy közös sablonnak megfeleltethetőek és könnyedén azonosíthatóak az URL közös sémájára kifejezve. Továbbá néhány félig-struktúrált adat lekérdező nyelvek, pl.: XQuery és HTQL, használhatóak HTML oldalak elemzésére és az adatok lekérdezésére és a tartalom átalakítására.

**4.3.4 DOM elemzés**

Önálló web böngésző beágyazásával (pl.: Internet Explorer, Mozilla) a programok irányítani tudják a böngészőt, hogy a kliens oldalon dinamikusan generált szkriptekből adatokat nyerjenek ki. Ezek a böngészők irányítják és elemzik a weboldalakat a DOM (Document Object Model) fában. Bizonyos programok le tudják kérdezni csak egy-egy részét az oldalaknak. Nyelvek, mint pl. az Xpath, használhatóak az eredmény DOM fa elemzésére.

**4.3.5 Függőleges felhalmozás**

Néhány vállalat kifejlesztett vertikum specifikus adatgyűjtési platformot. Ezek a platformok készítenek és figyelnek nagy mennyiségű botokat bizonyos különböző vertikumra ember nélkül és nem egy megjelölt céloldalra. Az előkészítés magában foglal egy tudásbázis felépítését az egész vertikumra és ezután a botok elkészülnek automatikusan. A platform robusztusságát a kinyert információk minőségével mérik (általában a mezők száma) és a skálázhatóságával (milyen gyorsan skálázható száz vagy ezernyi oldalakra). Ez a skálázhatóság gyakran az oldalak „hosszú farkát” (statisztikában az előfordulási grafikon domináló jobb oldalát) célozzák meg, a közös felhalmozást bonyolult vagy túl munkaigényes, hogy a tartalmat kibányásszák.

**4.3.6 Szemantikai annotáció felismerése**

A scrap-elt oldalak tartalmazhatnak metaadatokat, szemantikai jelöléseket és annotációkat, amik jelölhetnek bizonyos adattöredékeket. Ha az annotációk az oldalba vannak ágyazva (ahogy a Microformat-nál), akkor ez a technika a DOM elemzésnek egy különleges eseteként tekinthető. Más esetekben az annotációkat egy szemantikus rétegbe szervezik és ezeket a web oldalaktól külön tárolják és szervezik, tehát a scraper-ek le tudják kérdezni az adatok sémáját ebből a rétegből.

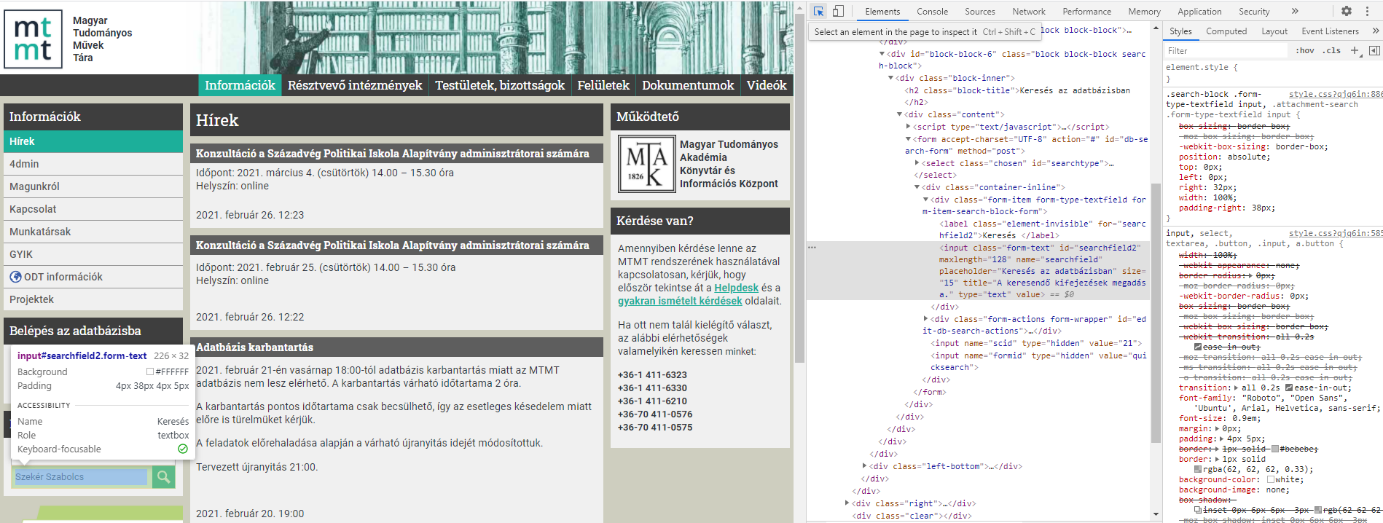
**4.3.7 Számítógépes látás – weboldal analízis**

A gépi tanulás és a számítógépes látás úgy próbálja meg felismerni és kinyerni az információt a web oldalakból, ahogy akár az ember tenné; vizuálisan értelmezve az oldalt.

**4.4 Selenium**

1. **Azonos feltételű kereső megvalósítása MTMT és Google Scholar rendszerekben**

Példa egy weboldal megfelelő részének megtalálásához a Chrome DevTools segítségével az MTMT oldalon:



1. **Adatok kigyűjtése, találatok megjelenítése**
2. **Találatok összehasonlítása és különbségek keresése**
3. **Automatikus/szoftverrel történő összehasonlítás előnye a manuálissal szemben**
4. **Adatok összehasonlítása és kiértékelése, különbségek felfedezése**
5. **Irodalomjegyzék**

<https://www.mtmt.hu/>

<https://scholar.google.com/>

<https://www.scraperapi.com/blog/best-google-scholar-apis-proxies/>

<https://en.wikipedia.org/wiki/Web_scraping>

<https://www.selenium.dev/>