Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи та інструменти видобутку веб-контенту

(тема)

Виконав:

студент II курсу, групи СПм-21-1

Філенко В.П.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ільїна І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Філенку Владиславу Петровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Методи та інструменти видобутку веб-контенту

затверджена наказом по університету від “ 07 ” листопада 2022 р. № 1454 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 13 грудня 2022 р.

3. Вхідні дані до роботи Мова програмування – Python 3.6. Бібліотеки – BeautifulSoup4, Selenium 3.14.0. Командна оболонка для обчислень – Jupyter Notebook.

4. Перелік питань, що потрібно опрацювати у роботі

1. Огляд ключових моментів галузі видобутку веб-контенту та data mining.
2. Дослідження документації, практичне опробування та порівняння відомих програмних засобів видобутку веб-контенту.
3. Огляд проблем автоматичного видобутку веб-контенту.
4. Програмна реалізація вирішення проблем автоматичного видобутку веб-контенту.
5. Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Слайд-презентація – 20 слайдів.

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд теоретичних відомостей про галузь	08.11.22-16.11.22	
	видобутку веб-контенту		
2	Вибір методики дослідження	17.11.22-19.11.22	
3	Дослідження програмних інструментів	20.11.22-22.11.22	
4	Виявлення та дослідження проблем автоматич-	23.11.22-25.11.22	
	ного видобутку веб-контенту		
5	Розробка програмних методів	26.11.22-30.11.22	
6	Оформлення матеріалів кваліфікаційної роботи	01.12.22-06.12.22	
7	Подання кваліфікаційної роботи керівникові та	07.12.22-08.12.22	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	09.12.22-12.12.22	
	вання		

Дата видачі завдання 07 листопада 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ільїна І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 120 с., 27 рис., 5 табл., 2 дод., 25 джерел.

ВИДОБУТОК ДАНИХ, ВИДОБУТОК ВЕБ-КОНТЕНТУ, ІНСТРУМЕНТИ ВИДОБУТКУ ВЕБ-КОНТЕНТУ, МЕРЕЖА, ВЕБ-САЙТ, ВЕБ-КОНТЕНТ, PYTHON.

Метою кваліфікаційної роботи є ознайомлення з галуззю data mining, огляд та порівняння методів та інструментів вилучення даних, а також представлення варіантів вирішення проблем, які можуть виникнути в процесі автоматичного видобутку знань з Інтернету.

У ході виконання роботи був проведений детальний огляд особливостей галузі видобутку даних, її основних положень та проблем, а також методів та інструментів вилучення інформації. Далі було наведено перелік відомих програмних засобів автоматичного видобутку даних, а також проведено їх порівняння. Також було представлено концепцію використання мови програмування Python і її бібліотек як інструменту усунення недоліків автоматичного видобутку знань. В практичній частині була проведена програмна реалізація вирішення проблем, які виникають під час автоматичного вилучення даних.

Програмна частина розроблялася з використанням мови програмування Python 3.6, бібліотек BeautifulSoup та Selenium.

ABSTRACT

Master’s thesis: 120 pages, 27 figures, 5 tables, 2 appendices, 25 sources.

DATA MINING, WEB CONTENT MINING, WEB CONTENT MINING TOOLS, NETWORK, WEBSITE, WEB CONTENT, PYTHON.

The purpose of the qualification work is to get acquainted with the field of data mining, to review and compare data extraction methods and tools, as well as to present options for solving problems that may arise in the process of automatically extracting knowledge from the Internet.

In the course of the work, a detailed review of the specifics of the field of data mining, its main priciples and problems, as well as methods and tools for information extraction were carried out. Next, a list of well-known software tools for automatic data extraction was given, as well as their comparison. The concept of using the Python programming language and its libraries as a tool to eliminate the shortcomings of automatic knowledge extraction was also presented. In the practical part, programming was implemented to solve problems that arise during automatic data extraction.

The programming part was developed using Python 3.6 programming language, BeautifulSoup and Selenium libraries.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML – мова розмітки гіпертексту (англ., HyperText Markup Language)

СУБД – система управління базами даних

ER-модель – модель «сутність-зв’язок» (англ., Entity-Relationship model)

SQL – мова структурованих запитів (англ., Structured Query Language)

OLAP – аналітична обробка у реальному часі (англ., OnLine Analytical Processing)

OEM – модель обміну об’єктами (англ., Object Exchange Model)

SKICAT – інструмент каталогізації та аналізу зображень неба (англ., Sky Image Cataloging and Analysis Tool)

URL – уніфікований локатор ресурсів (англ., Uniform Resource Locator)

IP – інтернет протокол (англ., Internet Protocol)

XML – розширювана мова розмітки (англ., Extensible Markup Language)

ISBN – міжнародний стандартний номер книги (англ., International Standard Book Number)

API – прикладний програмний інтерфейс (англ., Application Programming Interface)

CSS – каскадні таблиці стилів (англ., Cascading Style Sheets)

XPath – мова запитів XML-документів (англ., XML Path Language)

DOM – об'єктна модель документа (англ., Document Object Model)

xHTML – розширювана мова розмітки гіпертексту (англ., Extensible Hypertext Markup Language)

CMS – система керування вмістом (англ., Content Management System)

ВСТУП

Всесвітня павутина — це популярне та інтерактивне середовище з величезною кількістю доступних знань. Це сукупність документів, текстових файлів, зображень та інших форм даних з не завжди однорідною структурою. З огляду на вражаючий і непередбачуваний ріст інформації, доступної в Інтернеті, а також через неоднорідність і неструктурованість даних, пошук інформації став громіздким і трудомістким завданням. На сьогоднішній день це призводить до того, що більшість користувачів всесвітньої павутини стикаються з інформаційним перевантаженням. Веб-майнінг став рішенням цієї проблеми, використовуючи різноманітні методи видобутку даних для вилучення корисної інформації з Інтернету.

Об’єктом дослідження є видобуток веб-контенту, що є одним із видів веб-майнінгу.

Предметом дослідження є дослідження методів та інструментів видобутку веб-контенту.

Метою даної роботи є представлення концепції веб-видобутку, його методів та алгоритмів, а також огляд і порівняння різних видів інструментів видобутку веб-контенту.

Актуальність даної теми полягає в тому, що на сьогоднішній день видобуток веб-контенту застосовується в багатьох видах активності користувачів Інтернету. Наприклад, видобуток цього типу є дуже важливим інструментом адміністраторів різноманітних веб-ресурсів для моніторингу поведінки клієнтів, виходячи з аналізу вмісту їх веб-файлів, коментарів в соціальних мережах, блогах і на веб-форумах. Все це використовується для вилучення думки відвідувачів веб-ресурсів і аналізу їх настроїв. Цей вид майнінгу також використовується в освіті для покращення вилучення цінної інформації.

Практична цінність використання видобутку веб-контенту проявляється в

багатьох сферах застосування, таких як: удосконалення пошуку в пошукових системах; в електронній комерції; для збільшення прибутку з перехресних продаж; для помітного покращення якості обслуговування веб-сайтів; визначення потреб споживачів різних сфер послуг тощо. Насправді, застосування для веб-майнінгу можна знайти практично в усіх сферах діяльності користувачів в Інтернеті, обмежуючись лише фантазією.

Дослідження з обраної теми не несуть в собі новизни і являються оглядом і власною систематизацією вже існуючих праць, і можуть бути використані для подальшого розвитку в даній області.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність видобутку веб-контенту

За багато років свого існування, Інтернет перетворився на величезне сховище даних, і вилучення корисної інформації стало однією з найбільших проблем його користувачів. Пошукові системи вирішують цю проблему в більшості випадків, але вони, на жаль, не завжди демонструють високу точність знаходження даних. В основному, це пов’язано з тим, що ці дані можуть зустрічатися у вигляді різних форматів та форм. Значна частина інформації в Інтернеті є або напівструктурованою, або неструктурованою, і отримання потенційно корисних знань із цих різноманітних форматів останнім часом стало великою сферою досліджень.

Видобуток веб-контенту – це один із видів веб-майнінгу, який зосереджений на вилученні корисних шаблонів із доступного вмісту веб-документів. Пошуки в Інтернеті, з метою отримання абсолютно точної інформації, є складною задачею, а тому веб-майнінг – це техніка, яка використовується для швидкого отримання інформації з високим ступенем точності та надійності.

Даний розділ присвячений аналізу особливостей процесу веб-майнінгу, його видів, а також різних методів видобутку контенту, які застосовують до веб-документів.

1.2 Веб-майнінг: основні положення та класифікація

Веб-майнінг – це процес виявлення потенційно корисної та раніше невідомої інформації або знань з веб-даних [2]. Він використовується для отримання потрібної інформації, оцінки нових знань на основі відповідних даних, або для вивчення різних типів користувачів. Веб-майнінг включає в себе

методи інтелектуального аналізу для автоматичного виявлення та вилучення інформації з веб-документів і служб.

Для виявлення нових знань з величезної кількості даних, в минулому люди користувалися іншими методами, такими як ручний пошук та вилучення, статистичний аналіз, машинне навчання тощо. Порівнюючи ці методи з веб-майнінгом, інформаційний пошук в ньому працює автоматично, шляхом індексування тексту і вибору корисних знань, що забезпечується наявністю систем, в задачі яких входять пошук та вилучення інформації, її узагальнення, аналіз та перевірка [2].

Варто розглянути задачі веб-майнінгу детальніше [3]:

- пошук інформації/дослідження ресурсів;
- вилучення інформації;
- узагальнення;
- аналіз/перевірка.

Метою задачі пошуку інформації/дослідження ресурсів є автоматичне знаходження всіх релевантних документів, водночас відфільтровуючи нерелевантні. Пошукові системи є основним інструментом, який люди використовують для пошуку веб-інформації.

Побудова уніфікованих систем вилучення інформації є складною задачею, оскільки веб-контент є динамічним та різноманітним. Більшість таких систем використовують техніку «обгортки» для отримання конкретної інформації для конкретного сайту. Методи машинного навчання також використовуються для вивчення правил вилучення.

Метою завдання узагальнення є вивчення поведінки та інтересу користувачів. Тут використовуються такі методи інтелектуального аналізу даних, як правила кластеризації та асоціації.

Метою задачі аналізу/перевірки є отримання знань з даних, наданих з попередніх задач. На основі цих веб-даних можна створювати моделі для імітації та перевірки веб-інформації.

Веб-майнінг також поділяється на три категорії [3] (рисунок 1.1):

- видобуток веб-контенту (web content mining);
- видобуток веб-структур (web structure mining);
- аналіз використання веб-ресурсів (web usage mining).

Веб-контент – це видимі дані на веб-сторінках або будь-який тип інформації на просторах Інтернету, яка включає текст, аудіо, відео, зображення, HTML тощо. Процес вилучення різноманітних типів даних з веб-сторінок підпадає під категорію видобутку веб-контенту. Знання, які підлягають вилученню, можуть бути структурованими, напівструктурованими або неструктурованими.

Дана категорія веб-майнінгу перш за все асоціюється з текстовим аналізом у тому сенсі, що тут використовуються численні методи інтелектуального аналізу знань для дослідження вмісту веб-документів, і більшість даних, доступних у мережі, є у формі тексту. Також виконуються операції з набором даних, відомим як структуровані дані, і, на відміну від неструктурованих, вони зберігаються у формі таблиці.

Під видобутком веб-структур мають на увазі інструмент, який використовується для виявлення зв’язку між двома чи більше веб-сторінками, пов’язаними з необхідною інформацією. Основна мета аналізу веб-структур полягає в роботі зі структурою гіперпосилань на веб-сторінках. Це може бути зовнішньо- і внутрішньоструктурне існування, яке забезпечується за допомогою гіпертексту.

Принцип роботи даного інструменту полягає у використанні теорії графів із різними вузлами та посилання на з’єднання з усіма вузлами. Таким чином, можна представити зв’язок між різними веб-сторінками у формі веб-графа, у якому вузли веб-графа є веб-документами, а краї – це гіперпосилання, що існують між різними веб-сторінками. Це допомагає вилучити неідентифікований зв’язок або зв’язки між різними сторінками веб-сайту або різними сторінками різних веб-сайтів, що необхідно для покращення навігаційного шляху для його користувачів.

Як приклад використання видобутку веб-структур можна привести сферу бізнесу або електронної комерції, де можна створити групу користувачів, тобто кластери, для пошуку даних подібного типу в Інтернеті, що призводить до ефективного вдосконалення кількох видів діяльності та збільшення обсягів продажу.

Інтелектуальний аналіз використання веб-ресурсів − це процес, пов’язаний із отриманням будь-якої інформації з журналів сервера. Це інструмент, за допомогою якого можна легко провести аналіз уподобань користувачів в Інтернеті, тобто того, який тип даних їх цікавить. Наприклад, деякі користувачі зацікавлені в даних текстового типу, інші – в аудіо, відео або зображеннях.

Тому, для надання персоналізованого контенту для майбутнього клієнта, хорошою ідеєю було б виявити поведінку різних користувачів, які взаємодіють із веб-ресурсом. У файли журналу сервера різних веб-організацій кожного дня збирається великий обсяг даних. Очевидно, має сенс провести дослідження або отримати деякі корисні факти з цієї великої кількості знань, щоб зрозуміти потреби клієнтів, розробити схему перехресної маркетингової стратегії для різних послуг, що надаються тією чи іншою організацією, оцінити результативність різних рекламних кампаній, щоб представити більш цінні персоналізовані дані або вміст чи інформацію веб-користувачам і знайти найбільш відповідну логічно пов’язану структуру для їх веб-контенту. Таким чином можна виявити спільний інтерес групи користувачів, які взаємодіють з тим чи іншим ресурсом.

Використовуючи аналіз використання веб-сайтів, користувачі зможуть отримувати різні типи пропозицій від організацій, клієнтами яких вони є. Це може бути пошук нерухомості, сайти онлайн-покупок для певного продукту тощо. Пізніше ця інформація буде представлена у вигляді колекції сторінок, на які часто посилаються.

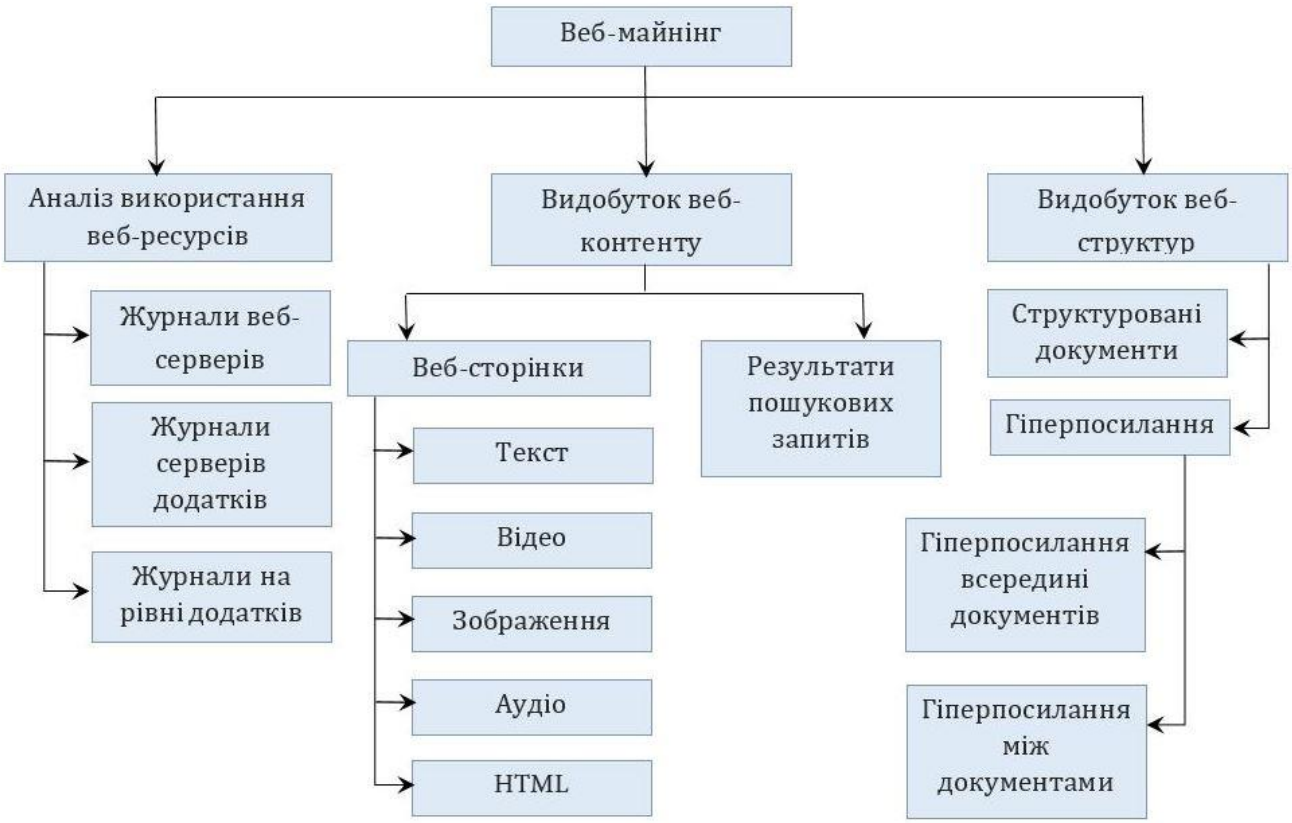


Рисунок 1.1 – Структура веб-майнінгу

Як було описано вище, видобуток веб-контенту − це процес знаходження корисної інформації всередині веб-документів. Під «веб-контентом» розуміють набір фактів, які були поміщені на веб-сторінку для передачі користувачам. Він може складатися з тексту, зображень, аудіо, відео, метаданих, гіперпосилань або структурованих записів, таких як списки та таблиці. Дослідження в галузі видобутку веб-контенту охоплюють виявлення ресурсів з Інтернету, категоризацію та кластеризацію документів, а також вилучення інформації з веб-сторінок.

Задачі веб-майнінгу можна розподілити по категоріям [4], як наведено в таблиці 1.1.

Таблиця 1.1 – Аналіз категорій веб-майнінгу

	Веб-майнінг		
	Видобуток веб-контенту	Аналіз використання веб-ресурсів	Видобуток веб-структур
Вигляд даних	Неструктуровані, напівструктуровані	Інтерактивність	Структура посилань
Основні дані	Текстовий документ, гіпертекстовий документ	Журнал сервера, браузера	Структура посилань
Представлення	Набір слів, термінів, фраз тощо	Таблиця відносин, граф	Граф
Метод	Машинне навчання, статистичний метод	Машинне навчання, асоціативні правила, статистичний метод	Власні алгоритми
Область застосування	Знаходження шаблонів в тексті	Побудова сайтів, менеджмент, маркетинг	Категоризація, кластерування

Так як тема даної роботи відноситься до особливостей видобутку веб-контенту, далі мова йтиме лише про дану категорію веб-майнінгу.

Процес видобутку веб-контенту відрізняється з двох різних точок зору: з точки зору отримання інформації та з точки зору бази даних [4]. Відмінності наведено в таблиці 1.2.

Таблиця 1.2 – Аналіз категорії «Видобуток веб-контенту»

	Видобуток веб-контенту	
	З точки зору отримання інформації	З точки зору бази даних
Вигляд даних	Неструктуровані, напівструктуровані	Напівструктурований веб-сайт як база даних
Основні дані	Текстовий документ, гіпертекстовий документ	Гіпертекстовий документ
Представлення	Реляційне, пакет слів, термінів, фраз тощо	Реляційне, граф з мітками ребер
Метод	Машинне навчання, статистичний метод	Асоціативні правила
Область застосування	Знаходження шаблонів в тексті	Знаходження частих підструктур, дослідження схеми веб-сайту

Узагальнені дослідження, які проведені в роботі [5], описують використання неструктурованих та напівструктурованих даних у режимі отримання інформації, і показують, що більшість досліджень використовують пакет слів, який базується на статистиці про окремі слова в ізоляції, для представлення неструктурованого тексту та беруть окреме слово, знайдене в навчальному наборі, як ознаку для пошуку. Для напівструктурованих даних усі дослідження використовують структури HTML всередині документів, а деякі використовували структури гіперпосилань між документами для представлення документів.

Стосовно точки зору бази даних, для кращого управління інформацією та запитами в Інтернеті майнінг завжди намагається визначити структуру веб-сайту, щоб перетворити його на базу даних. Інтелектуальний аналіз мультимедійних даних є частиною аналізу контенту, який спрямований на видобуток високорівневої інформації та знань із великих онлайнових мультимедійних джерел.

Підхід баз даних для веб-майнінгу намагається розробити методи організації напівструктурованих даних, що зберігаються в Інтернеті, у більш структуровані колекції інформаційних ресурсів. Тоді для аналізу цих колекцій можна використовувати стандартні механізми запитів до бази даних і методи аналізу інформації.

Дані для видобутку можуть існувати в різноманітних формах. Особливості видобутку даних різних типів, а також їх види описано в наступному підрозділі.

1.3 Типи даних, які можна видобувати

Як загальна технологія, видобуток даних може бути застосований до будь-якого типу знань, за умови, що вони мають значення для цільової програми. Найпростішими формами даних для data mining програм є дані бази даних, сховища даних і транзакцій [2]. Майнінг також можна застосовувати до

інших форм інформації (наприклад, потоків даних, упорядкованих/послідовних, графічних або мережевих, а також для просторових, текстових та мультимедійних даних), огляд яких буде представлено далі. Методи видобутку цих типів даних також будуть коротко представлені в наступних підрозділах. Видобуток даних, безперечно, продовжуватиме охоплювати нові їх типи у міру появи.

Реляційні бази даних є одним із найпоширеніших і найбагатших сховищ інформації, а отже, вони є однією з основних форм даних у дослідженнях data mining. Це забезпечується структурою СУБД, що містить в собі набір інструментів для керування та доступу до взаємопов'язаних знань.

Так як в більшість моделей баз даних представляють собою таблиці з атрибутами та рядками, ідентифікованими унікальним ключем і пов'язаними між собою (модель «сутність-зв'язок» (ER), як показано на рисунку 1.2), дані з них можна видобувати за допомогою запитів.

customer	(*cust_ID, name, address, age, occupation, annual_income, credit_information, category, . . .*)
item	(*item_ID, brand, category, type, price, place_made, supplier, cost, . . .*)
employee	(*empl_ID, name, category, group, salary, commission, . . .*)
branch	(*branch_ID, name, address, . . .*)
purchases	(*trans_ID, cust_ID, empl_ID, date, time, method_paid, amount*)
items_sold	(*trans_ID, item_ID, qty*)
works_at	(*empl_ID, branch_ID*)

Рисунок 1.2 – Зв'язки реляційної бази даних

Використовуючи запити, написані на мові реляційних запитів (наприклад, SQL), користувачі можуть отримувати з БД інформацію різного виду. За допомогою операцій, таких як об'єднання, вибір і проектування, можна вилучити інформація на кшталт: «список усіх товарів, які були продані за останній квартал». Використання агрегатних функцій, таких як sum (сума), avg (середнє), count (кількість), max (максимум) і min (мінімум), дає змогу робити більш складні запити, наприклад: «загальні продажі за останній місяць,

згруповані за галузями», «скільки транзакцій продажу відбулося в грудні?», або «у якого продавця були найбільші продажі?».

Використовуючи вищезазначену інформацію, можна шукати тенденції або шаблони даних [2]. Наприклад, системи видобутку знань можуть аналізувати дані клієнтів, щоб передбачити кредитний ризик нових клієнтів на основі їхнього доходу, віку та попередньої кредитної інформації. Системи вилучення знань також можуть виявляти відхилення, тобто товари з продажами, які далекі від очікуваних порівняно з попереднім роком. Такі відхилення можуть бути потім використані для подальшого аналізу та дослідження. Наприклад, системи видобутку даних можуть виявити зміни упаковки товару або значне підвищення ціни.

Видобуток даних зі сховищ даних дозволяє збирати інформацію, отриману з кількох джерел, і яка зберігається за єдиною схемою та зазвичай знаходиться на одному сайті. Знання в сховищах даних підлягають процесам очищення, інтеграції, перетворення, завантаження і періодичного оновлення, як показано на рисунку 1.3.

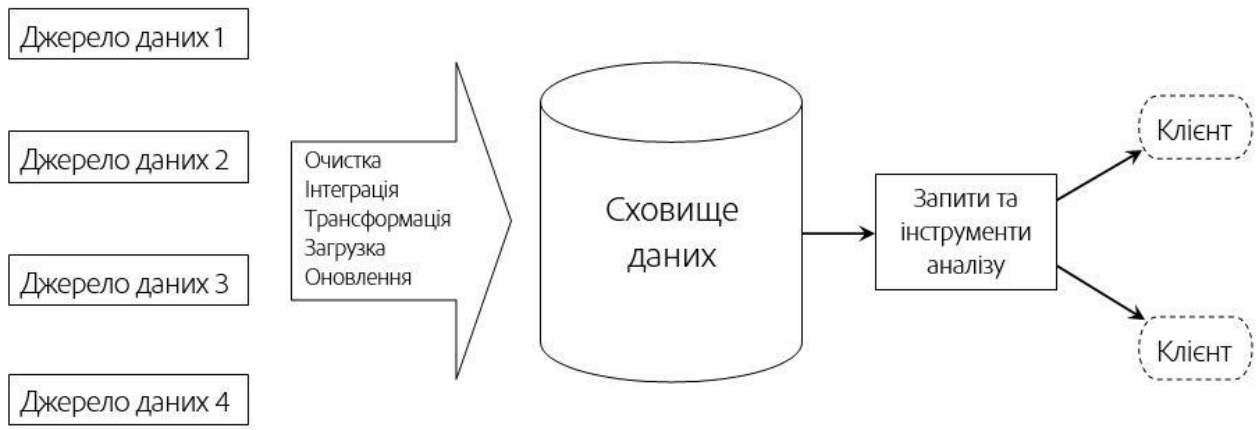


Рисунок 1.3 – Операції над даними в сховищі даних

Куб даних, за допомогою якого моделюються сховища даних, забезпечує

багатовимірне представлення інформації і попереднє обчислення та швидкий доступ до узагальнених знань, що дозволяє надавати підтримку OLAP [2]. Приклади операцій OLAP включають деталізацію та згортання, які дозволяють користувачеві переглядати дані з різним ступенем узагальнення. Наприклад, можна деталізувати факти про продажі, узагальнені за кварталами, щоб переглянути дані, узагальнені за місяцями. Так само можна згорнути факти про продажі, зведені за містами, щоб переглянути дані, зведені за країнами.

Хоча інструменти сховища даних допомагають підтримувати аналіз інформації, для поглибленого аналізу часто потрібні додаткові інструменти видобутку знань. Багатовимірний видобуток даних (також званий пошуковим багатовимірним видобутком даних) виконує видобуток знань в стилі OLAP. Тобто це дозволяє досліджувати численні комбінації вимірів на різних рівнях деталізації майнінгу інформації і, таким чином, має більший потенціал для виявлення цікавих моделей, що представляють знання.

Завдяки моніторингу транзакцій у транзакційній базі даних, що включає в себе відстежування покупок клієнта, елементів, що входять до неї, опису товару, даних про продавця чи філію тощо, користувачі можуть видобувати різного роду інформацію. Це може бути інформація на кшталт: «які товари добре продавалися разом?». Такий аналіз даних ринкового кошика дозволить об’єднувати групи товарів разом для здійснення стратегій збільшення продажів. Наприклад, знаючи, що принтери зазвичай купують разом із комп’ютерами, можна запропонувати певні принтери зі значною знижкою (або навіть безкоштовно) клієнтам, які купують певні комп’ютери, в надії продати більше комп’ютерів (які часто дорожчі за принтери).

Іншими словами, традиційні системи баз даних не в змозі проводити аналіз даних ринкового кошика. Але на щастя, за допомогою методів видобутку даних, проведених на даних транзакцій, це можна зробити шляхом видобутку частих наборів предметів, тобто наборів предметів, які часто продаються разом.

Окрім даних реляційної бази даних, даних сховища даних і даних транзакцій, існує багато інших видів даних, які мають різноманітні форми та

структури та досить різні семантичні значення [2]. Такі типи можна побачити в багатьох програмах: пов’язані з часом дані або дані послідовності (наприклад, історичні записи, дані фондової біржі, часових рядів і біологічної послідовності), потоки даних (наприклад, дані відеоспостереження та датчиків, які безперервно передаються), просторові дані (наприклад, карти), дані інженерного проектування (наприклад, проектування будівель, системних компонентів або інтегральних схем), гіпертекстові та мультимедійні дані (включно з текстом, зображеннями, відео та аудіо), графіки та мережеві дані (наприклад, соціальні та інформаційні мережі) і Інтернет.

Таке різноманіття створює все більше і більше проблем. Наприклад, як обробляти інформацію, що містить спеціальні структури (наприклад, послідовності, дерева, графіки та мережі) і конкретну семантику (наприклад, порядок, зображення, аудіо- та відеоконтент і зв’язок), а також як досліджувати шаблони, які несуть багату структуру та семантику.

З цих видів даних можна отримати різні види знань. Що стосується тимчасових даних, то можна видобувати банківські дані для змін тенденцій, що може допомогти в плануванні банківських кас відповідно до обсягу трафіку клієнтів. Дані фондової біржі можна отримати, щоб виявити тенденції, які можуть допомогти спланувати інвестиційні стратегії (наприклад, найкращий час для придбання акцій компаніями). Також можна досліджувати потоки даних комп’ютерної мережі, щоб виявляти вторгнення на основі аномалії потоків повідомлень, які можна виявити за допомогою кластеризації, динамічної побудови моделей потоків або порівняння поточних частих шаблонів із попередніми.

За допомогою просторових даних можна шукати шаблони, які описують зміни в рівнях бідності в мегаполісах залежно від відстані міста від основних магістралей. Зв’язки між набором просторових об’єктів можна перевірити, щоб виявити, які підмножини об’єктів є просторово автокорельованими або асоційованими. Шляхом аналізу текстових даних, таких як література про збір інформації за останні десять років, можна визначити еволюцію гарячих тем у

цій галузі. Досліджуючи коментарі користувачів щодо продуктів (які часто надсилаються у вигляді коротких текстових повідомлень), можна оцінити настрої клієнтів і зрозуміти, наскільки продукт сприймається ринком.

З мультимедійних даних можна видобувати зображення, щоб ідентифікувати об’єкти та класифікувати їх, призначаючи семантичні мітки або теги. Видобуваючи відеодані хокейної гри, можна виявити відеопослідовність, що відповідає голам. Веб-майнінг може допомогти дізнатися про розподіл інформації в мережі загалом, охарактеризувати та класифікувати веб-сторінки, а також розкрити веб-динаміку, асоціації та інші відносини між різними веб-сторінками, користувачами, спільнотами та веб-діяльністю.

Важливо мати на увазі, що в багатьох програмах присутні різні типи даних. Наприклад, у веб-майнінгу часто існують текстові та мультимедійні дані (наприклад, зображення та відео) на веб-сторінках, дані графіків, такі як веб-графіки, і картографічні дані на деяких веб-сайтах. У біоінформатиці геномні послідовності, біологічні мережі та тривимірні просторові структури геномів можуть співіснувати для певних біологічних об’єктів. Майнінг багатьох джерел складних даних часто призводить до плідних результатів завдяки взаємному покращенню та консолідації таких багатьох джерел. З іншого боку, це є складною задачею через труднощі в очищенні та інтеграції знань, а також через складну взаємодію між кількома джерелами таких даних.

Незважаючи на те, що такі дані вимагають складних засобів для ефективного зберігання, пошуку та оновлення, вони також створюють сприятливий ґрунт і викликають складні дослідження та проблеми впровадження методів вилучення інформації, тому видобуток таких даних є складною темою.

Видобуток веб-контенту пов'язаний з різноманітними проблемами на різних рівнях. Наступний підрозділ описує їх більш детально.

1.4 Проблеми видобутку даних

Видобуток даних – це динамічна галузь, що швидко розвивається, і має великі переваги. У цьому підрозділі будуть коротко окреслені основні проблеми дослідження цієї галузі, з розділенням їх на п’ять груп: методологія видобутку даних, взаємодія з користувачем, ефективність і масштабованість, різноманітність типів даних, а також видобуток даних і суспільство. Багато з цих проблем певною мірою розглядалися під час останніх досліджень і розробок інтелектуального аналізу даних і тепер вважаються стандартами; інші все ще знаходяться на стадії дослідження [2]. Проблеми продовжують стимулювати подальші дослідження та вдосконалення в галузі.

1.4.1 Методологія видобутку даних

Дослідники енергійно розробляють нові методології видобутку даних. Це включає в себе дослідження нових видів знань, видобуток у багатовимірному просторі, інтеграцію методів з інших дисциплін та розгляд семантичних зв’язків між об’єктами даних. Крім того, методології видобутку повинні враховувати такі проблеми, як невизначеність інформації, шум і неповноту. Деякі з методів досліджують, як визначені користувачем показники можна використовувати для оцінки цікавості виявлених шаблонів, а також для керування процесом дослідження. Варто детальніше розглянути різні аспекти методології видобутку даних [2].

Видобуток різних і нових видів знань: видобуток даних охоплює широкий спектр завдань аналізу і виявлення знань, від характеристики даних і розрізнення до аналізу асоціацій і кореляції, класифікації, регресії, кластеризації, аналізу послідовності, а також аналізу тенденцій і еволюції. Ці завдання можуть використовувати ту саму базу даних по-різному та потребують розробки численних методів видобутку даних. Завдяки різноманітності застосувань продовжують з’являтися нові завдання, що робить

майнінг динамічною та швидкозростаючою сферою. Наприклад, для ефективного виявлення знань в інформаційних мережах інтегрована кластеризація та ранжування можуть призвести до виявлення високоякісних кластерів і рангів об’єктів у великих мережах.

Видобуток знань у багатовимірному просторі: під час пошуку знань у великих наборах даних можливо досліджувати інформацію в багатовимірному просторі, тобто шукати цікаві моделі серед комбінацій вимірів (атрибутів) на різних рівнях абстракції. Такий видобуток відомий як багатовимірний (дослідницький) видобуток даних. У багатьох випадках дані можна агрегувати або розглядати у вигляді багатовимірного кубу. Інтелектуальний аналіз даних у просторі куба може значно підвищити потужність і гнучкість інтелектуального аналізу.

Видобуток даних − міждисциплінарне зусилля: потужність видобутку даних можна значно підвищити шляхом інтеграції нових методів із багатьох дисциплін. Наприклад, для вилучення даних із тексту, написаного природною мовою, має сенс об’єднати методи видобутку знань із методами пошуку інформації та обробки природної мови. Як інший приклад, можна розглянути виявлення програмних помилок у великих програмах. Ця форма видобутку інформації, відома як видобуток помилок, отримує переваги від включення знань інженерії програмного забезпечення в процес видобутку даних.

Підвищення можливостей виявлення в мережевому середовищі: більшість об’єктів даних знаходяться у зв’язаному або взаємопов’язаному середовищі, будь то Інтернет, зв’язки з базами даних, файли чи документи. Семантичні зв’язки між кількома об’єктами можна використовувати з перевагою: знання, отримані в одному наборі об’єктів, можна використовувати для прискорення відкриття знань у пов’язаному або семантично пов’язаному наборі об’єктів.

Обробка невизначеності, шуму чи неповноти даних: дані часто містять шум, помилки, винятки чи невизначеність або є неповними. Помилки та шум можуть заплутати процес видобутку знань, що призведе до виведення

помилкових шаблонів. Очищення даних, попередня їх обробка, виявлення та видалення залишків і обґрунтування невизначеності є прикладами методів, які необхідно інтегрувати з процесом видобутку даних.

Оцінка шаблонів і інтелектуальний аналіз, керований шаблонами чи обмеженнями: не всі шаблони, створені процесами видобутку даних, цікаві. Від користувача залежить те, що робить шаблон цікавим. Тому необхідні методи оцінки цікавості виявлених закономірностей основані на суб'єктивних показниках. Вони оцінюють цінність шаблонів щодо певного класу користувачів на основі їх переконань або очікувань. Більше того, використовуючи показники цікавості або визначені користувачем обмеження для керування процесом виявлення, можна створити більш цікаві шаблони та зменшити простір пошуку.

1.4.2 Взаємодія користувача

Користувач відіграє важливу роль у процесі видобутку знань. Серед цікавих областей дослідження — як взаємодіяти із системою вилучення даних, як залучити базові знання користувача до майнінгу та як візуалізувати й зрозуміти результати видобутку. Нижче представлено кожне з вещепереліченого [2].

Інтерактивний видобуток даних: процес видобутку даних має бути високоінтерактивним. Таким чином, важливо створювати гнучкі інтерфейси користувача та дослідницьке середовище майнінгу, що полегшує взаємодію користувача з системою. Наприклад, де користувач зможе спочатку взяти вибірку набору даних, дослідити їх загальні характеристики і оцінити потенційні результати видобутку. Інтерактивний майнінг повинен дозволяти користувачам динамічно змінювати фокус пошуку, уточнювати запити на основі повернутих результатів, а також інтерактивно вивчати, розрізати та повертати простір даних і знань, динамічно досліджуючи «простір куба» під час майнінгу.

Включення фонових знань: базові знання, обмеження, правила та інша інформація щодо досліджуваної області має бути включена в процес виявлення знань. Такі знання можуть бути використані для оцінки шаблонів, а також для спрямування пошуку до цікавих закономірностей.

Спеціальний видобуток даних і мови запитів видобутку даних: мови запитів (наприклад, SQL) відіграють важливу роль у гнучкому пошуку, оскільки вони дозволяють користувачам створювати однорічні запити. Так само високорівневі мови запитів видобутку даних або інші високорівневі гнучкі інтерфейси користувача надають користувачам свободу визначати спеціальні завдання вилучення знань. Це має полегшити специфікацію відповідних наборів даних для аналізу, знань домену, типів знань для видобутку, а також умови та обмеження, які повинні застосовуватися до виявлених шаблонів. Оптимізація обробки таких гнучких запитів на майнінг є ще одним перспективним напрямком дослідження.

Презентація та візуалізація результатів видобутку даних: це те, як система вилучення знань може яскраво та гнучко представити результати видобутку даних, щоб виявлені знання могли бути легко зрозумілими та безпосередньо використаними людьми. Це особливо важливо, якщо процес аналізу даних є інтерактивним. Така презентація вимагає, щоб система використовувала виразні представлення знань, зручні для користувача інтерфейси та методи візуалізації.

1.4.3 Ефективність і масштабованість

Під час порівняння алгоритмів видобутку даних завжди враховуються ефективність і масштабованість [2]. Оскільки кількість даних в Інтернеті продовжує зростати, ці два фактори є особливо критичними.

Ефективність і масштабованість алгоритмів видобутку даних: алгоритми вилучення знань мають бути ефективними та масштабованими, щоб ефективно отримувати інформацію з величезних обсягів даних у багатьох сховищах або в

динамічних потоках даних. Іншими словами, час роботи алгоритму видобутку знань має бути передбачуваним, коротким і прийнятним для програм. Ефективність, масштабованість, продуктивність, оптимізація та можливість виконання в режимі реального часу є ключовими критеріями, які стимулюють розробку багатьох нових алгоритмів обробки інформації.

Алгоритми паралельного, розподіленого та інкрементального видобутку даних: величезний розмір багатьох наборів даних, широкий їх розподіл і обчислювальна складність деяких методів є факторами, які спонукають до розробки алгоритмів паралельного та розподіленого видобутку. Такі алгоритми спочатку розбивають дані на «шматочки». Кожна деталь потім обробляється паралельно шляхом пошуку закономірностей. Паралельні процеси можуть взаємодіяти один з одним, а шаблони з кожного розділу згодом об’єднуються.

Хмарні обчислення та кластерні обчислення, які використовують комп’ютери розподіленим і спільним способом для вирішення дуже великомасштабних обчислювальних завдань, також є активними темами дослідження паралельного аналізу даних. Крім того, висока вартість деяких процесів видобутку знань і інкрементний характер введення сприяють інкрементальному видобутку інформації, який включає нові оновлення даних без необхідності видобувати їх «з нуля». Такі методи виконують поступову модифікацію знань, щоб виправити та зміцнити те, що було відкрито раніше.

1.4.4 Різноманітність типів баз даних

Широке розмаїття типів баз даних створює проблеми для аналізу інформації [2], про які йтиме мова надалі.

Робота зі складними типами даних: різноманітні програми генерують широкий спектр нових типів даних, від структурованих, таких як реляційні дані та дані сховища даних, до напівструктурованих і неструктурованих; від стабільних сховищ до динамічних потоків даних; від простих об’єктів даних до часових, біологічних послідовностей, даних датчиків, просторових даних,

гіпертекстових та мультимедійних даних, коду програмного забезпечення, веб-даних з різних соціальних мереж. Нереалістично очікувати, що одна система майнінгу зможе видобувати всі види знань, враховуючи їх різноманітність і наявність різних цілей видобутку. Системи вилучення даних, призначені для домену або програми, створюються для глибокого аналізу конкретних типів знань. Створення ефективних інструментів вилучення інформації для різноманітних застосувань залишається складною та активною областю досліджень.

Динамічні, мережеві та глобальні сховища даних для майнінгу: численні джерела даних об’єднані Інтернетом і різними видами мереж, утворюючи гігантські, розподілені та неоднорідні глобальні інформаційні системи та мережі. Виявлення знань із різних джерел структурованих, напівструктурованих або неструктурованих, але взаємопов’язаних даних із різноманітною семантикою створює великі проблеми для обробки даних. Видобуток таких гігантських взаємопов’язаних інформаційних мереж може допомогти розкрити набагато більше закономірностей і знань у гетерогенних наборах даних, ніж можна виявити з невеликого набору ізольованих сховищ. Веб-майнінг, багатоджерельний видобуток інформації і майнінг інформаційних мереж стали складними галузями вилучення знань, які зараз стрімко розвиваються.

1.4.5 Видобуток даних і суспільство

Варто розглянути питання про те, як аналіз даних впливає на суспільство; кроки, якими користуються алгоритми видобутку даних, щоб зберегти конфіденційність людей; чи використовують люди видобуток даних у повсякденному житті, навіть не підозрюючи про це? Ці питання піднімають багато проблем [2].

Соціальний вплив видобутку даних: оскільки видобуток даних проникає в наше повсякденне життя, важливо вивчати його вплив на суспільство. Варто

визначити, як можна використовувати технологію вилучення знань на благо суспільства і як можна захиститися від його неправильного використання. Неналежне розкриття або використання даних, потенційне порушення прав особи на конфіденційність і захист інформації є проблемними питаннями, які потребують вирішення.

Видобуток даних із збереженням конфіденційності: видобуток даних допомагає науковим відкриттям, управлінню бізнесом, відновленню економіки та захисту безпеки (наприклад, виявлення зловмисників і кібератак у реальному часі). Однак він створює ризик розголошення особистої інформації людини. Філософія концепції публікації даних із збереженням конфіденційності полягає в дотриманні конфіденційності інформації і збереженні анонімності людей під час успішного майнінгу.

Невидимий видобуток даних: не можна очікувати, що всі в суспільстві навчаться та опанують методи видобутку знань. Все більше і більше систем повинні мати вбудовані функції майнінгу, щоб люди могли виконувати вилучення даних або використовувати результати вилучення, просто клацаючи мишкою, без будь-яких знань про необхідні для цього алгоритми. Інтелектуальні пошукові системи та інтернет-магазини виконують такий невидимий аналіз даних, включаючи його у свої компоненти для покращення їх функціональності та продуктивності. Часто це робиться без відома самого користувача. Наприклад, купуючи товари в Інтернеті, користувачі можуть не знати, що магазин, ймовірно, збирає дані про моделі купівлі своїх клієнтів, які можуть бути використані, щоб рекомендувати інші товари для покупки в майбутньому.

Визначившись з тим, які типи даних можна видобувати, варто перейти до огляду методів та інструментів вилучення інформації. Для видобутку веб-контенту застосовуються різні методи та інструменти отримання даних, описані нижче.

1.5 Методи та інструменти видобутку веб-контенту

Видобуток веб-контенту має такі підходи до вилучення даних: неструктурований видобуток, структурований видобуток, напівструктурований видобуток і мультимедійний видобуток [4]. Веб-сторінка складається з тексту, зображень, аудіо, відео, метаданих, гіперпосилань або структурованих записів і таблиць. Це все можна класифікувати як неструктуровані, структуровані, напівструктуровані та мультимедійні дані. Методи інтелектуального аналізу, які використовуються для виявлення знань, — це інтелектуальний аналіз неструктурованого тексту, інтелектуальний аналіз структурованих даних, аналіз напівструктурованих даних і інтелектуальний аналіз мультимедійних даних, як наведено в таблиці 1.3.

Таблиця 1.3 – Типи даних, методи та інструменти видобутку веб-контенту

Тип даних веб-контенту	Метод видобутку	Інструменти
Неструктуровані дані	Видобуток неструктурованих даних	Видобуток інформації
		Відстеження тем
		Резюмування
		Категоризація
		Кластерування
		Візуалізація інформації
Структуровані дані	Видобуток структурованих даних	Видобуток контенту сторінки
		Пошуковий робот
		Врапер (Wrapper Generator)
Напівструктуровані дані	Видобуток напівструктурованих даних	Модель обміну об'єктами (OEM)
		Видобуток зверху вниз
		Мова вилучення веб-даних
Дані мультимедіа	Видобуток даних мультимедіа	SKICAT
		Мультимедійний майнер
		Відповідність кольорової гістограми
		Виявлення границі

Ознайомившись з типами даних веб-контенту, тепер варто детальніше розглянути різноманітні методи та інструменти його видобутку.

1.5.1 Видобуток неструктурованих даних

Текстовий документ є формою неструктурованих даних. Більшість даних, які доступні в Інтернеті, є неструктурованими. Дослідження застосування методів видобутку даних до неструктурованих даних відоме як виявлення знань у текстах.

Для видобутку неструктурованих даних використовуються наступні інструменти [6]:

- видобуток інформації (information extraction);
- відстеження тем (topic tracking);
- резюмування (summarization);
- категоризація (categorization);
- кластерування (clustering);
- візуалізація інформації (information visualization).

Для вилучення інформації з неструктурованих даних, яка присутня в мережі, використовується зіставлення зразків. Принцип роботи цього алгоритму полягає у відстеженні ключових слів та фраз, і подальшому знаходженні зв’язків ключових слів у тексті. Цей метод дуже корисний для аналізу великих обсягів строкових даних. Вилучення інформації перетворює неструктурований текст у більш структуровану форму. Спочатку з вилучених даних видобувається інформація, а потім за допомогою різних типів правил знаходяться упущена дані. Вилучення інформації, що робить невірні прогнози щодо даних, відкидається.

Техніка відстеження тем перевіряє документи, які переглядає користувач, і вивчає його профіль. Вона відбирає документи, відповідно до інтересів користувачів. Відстеження тем застосовується у пошуковій системі yahoo, де користувач вказує ключове слово, і якщо з’являється щось, пов’язане з цим

словом, користувач отримує про це повідомлення. Цю техніку можна застосовувати в багатьох областях. Наприклад, у сферах медицини та освіти. У галузі медицини лікарі легко дізнаються про новітні методи лікування. У сфері освіти вона використовується, щоб знайти найновішу довідку для дослідницької роботи. Недоліком техніки є те, що в результаті пошуку потрібної теми, можна отримати непотрібну інформацію.

Техніка резюмування використовується для зменшення розміру вмісту документа шляхом збереження ключових точок. Це допомагає користувачеві вирішити, читати тему чи ні. Час, витрачений технікою на резюмування документа, менше часу, який витрачається користувачем на читання першого абзацу. Дана техніка використовує два методи: екстракційний метод і абстрактний метод. Екстракційний метод вибирає підмножину фраз, речень і слів для формування резюме з оригінального тексту. Абстрактний метод створює внутрішнє семантичне представлення, а потім використовує техніку створення природної мови для створення резюме. Це резюме може містити слова, яких немає в оригінальному документі.

Екстрактивні методи резюмування використовуються переважно у порівнянні з абстрактними методами. Вони шукають заголовки та підзаголовки, щоб знайти важливі моменти цього документа. Техніка резюмування може працювати разом із технікою відстеженням тем.

Техніка категорозації визначає основну тему шляхом розміщення документів у попередньо визначеній групі. Техніка підраховує кількість слів у документі, і це визначає основну тему. Відповідно до теми документу надається ранг. Перше місце займають документи, більшість яких містить певну тему. Дана техніка допомагає надавати клієнтську підтримку галузям і бізнесу.

Техніка кластерування використовується для групування схожих документів. У цьому групуванні документів не здійснюється на основі заздалегідь визначених тем, а робиться на ходу. Деякі документи можуть відображатися в іншій групі. У результаті корисні документи не виключаються з результатів пошуку. Ця техніка допомагає користувачеві вибрати тему, яка

його цікавить.

Візуалізація використовує виділення ознак та індексування ключових термінів. Документи, що схожі між собою, виявляються шляхом візуалізації. Великі текстові матеріали представлені тут у вигляді візуальних карт або ієрархії, де дозволено можливість перегляду, що допомагає візуально аналізувати вміст. Також користувач може взаємодіяти шляхом збільшення розмірів, масштабування та створення підкарт графіків.

1.5.2 Видобуток структурованих даних

Структуровані дані зазвичай являють собою записи даних, отримані з основної бази даних і відображені на веб-сторінках. Їх можна відобразити у вигляді таблиць або форм. Інформацію з цих джерел можна отримати за допомогою методів, які застосовують для вилучення структурованих даних. Це може бути корисним у наданні цінних послуг шляхом збору інформації з різних джерел, наприклад, індивідуальний збір веб-інформації, порівняння покупок, мета-пошук тощо.

Для видобутку структурованих даних використовуються наступні інструменти [7]:

- видобуток контенту сторінки (page content mining);
- пошуковий робот (web crawler);
- врапер (wrapper generator).

Видобуток контенту сторінки – це техніка, яка використовується для вилучення структурованих даних, які працюють на сторінках, які ранжуються традиційними пошуковими системами. Сторінки класифікуються шляхом порівняння рейтингу їх вмісту.

Web Crawler — це відносно проста автоматизована програма або сценарій, який методично сканує сторінки Інтернету для створення індексу необхідних даних. Ці програми зазвичай призначені для одноразового використання, але їх можна запрограмувати і на тривале використання. Існує

кілька способів використання цих роботів, мабуть, найпопулярнішим є пошукові системи, які використовують їх для надання користувачам Інтернету потрібних веб-сайтів. Інші типи користувачів включають в себе лінгвістів і дослідників ринку, або будь-кого, хто намагається організовано шукати інформацію в Інтернеті. Альтернативами назви «пошуковий робот» є: веб-сканер, веб-павук, веб-робот, бот, сканер, автоматичний індексатор тощо. Програми сканера можна придбати в Інтернеті або в багатьох компаніях, які продають комп'ютерне програмне забезпечення, і їх можна завантажити на більшість комп'ютерів.

Існують різні способи використання веб-сканерів, але, по суті, веб-сканером може користуватися кожен, хто прагне збирати інформацію в Інтернеті. Пошукові системи часто використовують веб-сканери для збору інформації про те, що доступно на загальнодоступних веб-сторінках. Їхня головна мета — збирати дані, щоб користувачі Інтернету, вводячи пошуковий термін на своєму сайті, могли швидко надавати користувачам відповідні веб-сайти. Лінгвісти можуть використовувати веб-сканер для виконання текстового аналізу; тобто вони можуть прочесати Інтернет, щоб визначити, які слова зазвичай вживаються сьогодні. Дослідники ринку можуть використовувати веб-сканер, щоб визначити й оцінити тенденції на певному ринку.

Для полегшення ефективності пошуку у Всесвітній павутині, було створено кілька метапошукових систем, які не здійснюють пошук самостійно, а використовують доступні пошукові системи для знаходження потрібної інформації. Метапошукові системи підключаються до пошукових систем за допомогою враперів. Для кожної пошукової системи, підключеної до неї, є оболонка, яка перекладає запит користувача на рідну мову запиту та формат системи. Врапер також витягує відповідну інформацію зі сторінки результатів HTML пошукової системи [9].

1.5.3 Видобуток напівструктурованих даних

Напівструктуровані дані виникають, коли джерело не накладає на дані жорсткої структури, і якщо виникає потреба отримати інформацію з веб-сторінки та заповнити базу даних. Також вони виникають, коли знання поєднуються з кількох різнорідних джерел, наприклад, бібліографічні дані, де деякі книги написані одним автором, а деякі – двома чи більше авторами.

Веб-сторінки мають певну притаманну структуру, яку можна легко розпізнати, але одна веб-сторінка може суттєво відрізнятися від іншої, тому говорять, що дані веб-сторінки є напівструктурованими. У випадку структурованих даних, їх можна легко вилучити, надсилаючи запити, але в такому випадку важко робити запит на вилучення текстових даних. Щоб їх витягти, потрібен якийсь опис того, що має бути видобуто.

Для вилучення напівструктурованих даних можна застосувати такі методи [7]:

- модель обміну об'єктами (OEM);
- видобуток зверху вниз;
- мова вилучення веб-даних.

Необхідна інформація витягується з напівструктурованих даних і збирається в групу корисної інформації, а потім зберігається в моделі обміну об’єктами (OEM). Це допомагає користувачеві точно зрозуміти структуру інформації, доступної в Інтернеті, і найкраще підходить для неоднорідного та динамічного середовища. Основною особливістю моделі обміну об'єктами є самоопис, адже в такому випадку немає необхідності описувати заздалегідь структуру об'єкта.

Техніка видобування зверху вниз зосереджена на вилученні складних об’єктів із веб-джерел і розкладанні їх на менш складні об’єкти, доки не буде вилучено атомарні. За допомогою цієї техніки достатньо лише кількох зразків для вилучення сотень об’єктів на нову веб-сторінку. Основна ідея цього підходу полягає в пошуку об'єктів, ідентичних тому, який розглядається. Він

надається користувачем, і це дуже важливо, оскільки вся процедура вилучення залежить від коректності зразка об’єкта.

Стратегія «зверху вниз» працює шляхом обходу структури об’єкта-зразка у формі попереднього розташування, огляду всіх його компонентів і об’єднання їх для формування нового кінцевого об’єкта. Кожен новий об’єкт розпізнається та виділяється повністю до ідентифікації його складових.

Техніка мови вилучення веб-даних перетворює веб-дані на структуровані дані. Потім вони доставляються кінцевим користувачам. Дані зберігаються у вигляді таблиць.

1.5.4 Видобуток даних мультимедіа

Видобуток мультимедійних даних стосується видобутку мультимедійного контенту. Іншими словами, це вивчення великих обсягів мультимедійної інформації з метою пошуку закономірностей або статистичних зв’язків [8]. Коли дані зібрані, комп’ютерні програми використовуються для їх аналізу та пошуку значущих зв’язків. Ця інформація може бути використана в маркетингу для виявлення споживчих звичок. Але в основному вона використовується урядами для вдосконалення соціальних систем. Видобуток мультимедійних даних спрямований на виявлення закономірностей, вилучення правил і стосується отримання знань із аналізу мультимедійних баз даних, зокрема, різноманітних аспектів.

Основним реквізитом видобутку мультимедійних даних є збір величезних обсягів інформації. Ключовим фактором є розмір вибірки під час аналізу даних, оскільки прогнозовані тенденції та закономірності, швидше за все, будуть неточними з меншою вибіркою. Ці знання можна зібрати з різних носіїв, включаючи відео, звукові файли та зображення. Деякі експерти також вважають просторові дані та текст одним із видів мультимедіа.

Обробка зображень існує вже досить давно. Вона зосереджена на виявленні незвичайних шаблонів, а також на вилученні зображень. Видобуток

зображень пов’язує різні зображення у великій базі даних.

Видобуток відеоданих складніший, ніж видобуток зображень, оскільки тут йде мова про колекцію рухомих зображень у формі анімації. Видобуток відео передбачає пошук зв’язків між відеокліпами та виявлення в них незвичайних шаблонів.

Аудіодані складаються з радіо, мовлення або розмовної мови. Щоб отримати аудіодані, їх можна спочатку перетворити на текст за допомогою методів транскрипції мовлення, а потім отримати текстові дані. Їх також можна отримати безпосередньо за допомогою методів обробки аудіоінформації, а потім видобути вибрані аудіокліпи.

Для вилучення даних мультимедіа застосовують такі методи:

- SKICAT;
- мультимедійний майнер;
- відповідність кольорової гістограми;
- виявлення границі.

SKICAT — успішна система аналізу та каталогізації астрономічних даних, яка створює цифровий каталог об’єктів неба. Вона використовує техніку машинного навчання для перетворення цих об’єктів на класи, доступні для розуміння людині. SKICAT включає в себе техніку обробки зображень і класифікації даних, що допомагає класифікувати дуже великий набір класифікацій.

Мультимедійний майнер складається з чотирьох основних етапів: екскаватор зображень для вилучення зображень і відео; препроцесор для вилучення функцій зображення, які зберігаються в базі даних; ядро пошуку для зіставлення запитів із зображенням і відео, доступними в базі даних; модуль дослідження, який виконує підпрограми аналізу інформації про зображення, щоб відстежити в них шаблони.

Зіставлення кольорової гістограми складається з вирівнювання кольорової гістограми та згладжування. Вирівнювання намагається знайти кореляцію між компонентами кольору. Проблема, з якою стикається

вирівнювання, полягає в проблемі розріджених даних, яка полягає в наявності небажаних артефактів у вирівняних зображеннях. Ця проблема вирішується за допомогою згладжування.

Техніка виявлення границі автоматично визначає границі між кадрами у відео.

1.6 Висновки

У даному розділі був проведений огляд способів видобутку даних, доступних в Інтернеті. Також були детально описані різні види веб-майнінгу у формах видобутку веб-контенту, веб-структур та використання веб-ресурсів. Далі були розглянуті види даних для видобутку і різноманітні проблеми, з якими зустрічаються користувачі в процесі вилучення інформації з мережі. Крім того, у розділі було приділено велику увагу процесу видобутку веб-контенту, а також методам та технікам вилучення даних, що зберігаються у веб-джерелах.

Як висновок, можна сказати, що майнінг веб-контенту є корисним інструментом вирішення проблем пошуку інформації та задоволення потреб користувачів, відповідно до їх інтересів. Він є незамінним у сферах програм електронної комерції, ділового світу, соціальних мереж тощо. Проблеми, пов’язані з пошуком потрібної інформації за допомогою пошукових систем, можуть бути вирішені різноманітними методами видобутку, якщо вони використовуються точно відповідно до вимог користувача. Незважаючи на те, що доступно багато різних методів для видобутку різноманітних типів даних у мережі, існує потреба в подальшому покращенні ефективності та результативності отримання потрібної інформації з мережі.

У наступному розділі буде проведено детальне ознайомлення з різними програмними засобами видобутку веб-контенту.

2 ОГЛЯД ТА ПОРІВНЯННЯ ПРОГРАМНИХ ІНСТРУМЕНТІВ ВИДОБУТКУ ВЕБ-КОНТЕНТУ

2.1 Програмні засоби видобутку веб-контенту

Програмні засоби видобутку веб-контенту можуть отримувати необхідну та корисну інформацію з доступних даних у мережі. Зазвичай це робиться шляхом скачування цих даних у різних формах. Очевидно, що через їх різноманітність та неоднорідність в Інтернеті, цей процес може бути дуже трудомістким і виснажливим, і тому, з метою спрощення процесу отримання потрібної інформації, використовуються спеціальні інструменти. В широкому сенсі, їх можна розділити на комерційні та некомерційні.

З огляду на всі доступні інструменти здається, що всі вони задовольняють потреби користувачів в один і той самий спосіб. Дослідження [10] показують, що з 2014 року було створено близько 238 різних типів інструментів зі схожою функціональністю, а в деяких випадках і різними функціями. Однак більшість із них мають відмінні особливості у порівнянні один з одним. Далі у цьому розділі буде проведено ознайомлення з популярними інструментами видобутку веб-контенту, а також будуть проведені оцінка та порівняльний аналіз їх відмінностей та переваг.

2.2 Відмінності між пошуковими роботами та вебскраперами

Програми для видобутку веб-контенту поділяються на два види: пошукові роботи (Web Crawlers) та вебскрапери (Web Scrapers). Варто розглянути детальніше відмінності цих двох видів програмних засобів, перед тим як ознайомитись з популярними інструментами веб-майнінгу [11]. Це допоможе потенціальному користувачу узгодити власний варіант використання з правильною методологією збору даних, а також зрозуміти основні переваги та

проблеми кожного варіанту.

Пошуковий робот – це програма, що використовується для індексування інформації на веб-сторінках за допомогою ботів, також відомих як сканери. По суті, вони виконують ту ж роботу, що і пошукові системи, тобто переглядають сторінки в цілому та проводять їх індексування. Коли бот сканує веб-сайт, він переглядає кожну сторінку та кожне посилання до останнього рядка, вишукуючи будь-яку інформацію.

Пошукові роботи в основному використовуються основними пошуковими системами, такими як Google, Yahoo, Bing, статистичними агентствами та великими онлайн-агрегаторами. Процес веб-сканування пошуковим роботом зазвичай фіксує загальну інформацію, тоді як вебскрапер зосереджується на конкретних фрагментах набору даних.

Вебскрапери схожі на пошукових роботів тим, що вони намагаються ідентифікувати конкретно задані дані на веб-сторінках [11]. Ключова відмінність полягає в тому, що у випадку вебскрапінгу користувач знає точний ідентифікатор набору даних, наприклад, структуру елементів HTML для веб-сторінок, з яких потрібно витягти дані.

Вебскрапінг − це автоматизований спосіб вилучення певних наборів даних, іншими словами «вишкрібання» (від англ. scraping) інформації за допомогою ботів. Після збору потрібної інформації її можна використовувати для порівняння, перевірки та аналізу на основі потреб і цілей конкретного бізнесу.

Є декілька популярних способів, в яких компанії використовують вебскрапінг для досягнення своїх бізнес-цілей [11]:

- дослідження;
- роздрібна торгівля/електронна комерція;
- захист бренду.

Дослідження: дані часто є невід’ємною частиною будь-якого дослідницького проекту, незалежно від того, чи є він суто академічним за своєю природою, чи використовується для маркетингових, фінансових або

інших бізнес-додатків. Можливість збирати дані користувачів у режимі реального часу та визначати моделі поведінки, наприклад, може мати першочергове значення при спробі зупинити глобальну пандемію або визначити конкретну цільову аудиторію.

Роздрібна торгівля/електронна комерція: компаніям, особливо в сфері електронних комунікацій, необхідно регулярно проводити аналіз ринку для підтримки переваги. Релевантні набори даних, які збирають як зовнішні, так і внутрішні роздрібні підприємства, включають ціни, аналіз, асортимент, спеціальні пропозиції тощо.

Захист бренду: збір даних стає невід'ємною частиною захисту від шахрайських дій з брендом, а також виявлення зловмисників, які незаконно отримують прибуток від корпоративної інтелектуальної власності (назви, логотипи, відтворення елементів продукції). Збір даних допомагає компаніям відстежувати, ідентифікувати та вживати заходів проти таких кіберзлочинців.

Основними перевагами вебскрапінгу є висока точність, економічність та точне визначення.

Висока точність: вебскрапери допомагають уникати людських помилок при проведенні різних операцій, для впевненості, що інформація, яку отримує користувач, є на 100% точною.

Економічність: вебскрапінг може бути економічно ефективнішим, оскільки найчастіше для виконання роботи знадобиться менше персоналу, і в багатьох випадках можна отримати доступ до повністю автоматизованого рішення, яке не потребує жодної інфраструктури з боку користувача.

Точне визначення: багато вебскраперів дозволяють фільтрувати саме ті точки даних, які шукає користувач, тобто він може вирішити, що для конкретної роботи буде збирати, наприклад, зображення, а не відео; ціни, а не описи. Це може допомогти заощадити час, пропускну здатність і гроші в довгостроковій перспективі.

Щодо пошукових роботів, їх основними перевагами є [11]:

- глибоке занурення;

- робота у режимі реального часу;
- гарантія якості.

Метод глибокого занурення передбачає поглиблену індексацію кожної цільової сторінки. Це може бути корисним, наприклад, при розкритті та зборі інформації в глибині Всесвітньої павутини.

При роботі у режимі реального часу веб-сканування є кращим рішенням для компаній, яким потрібен знімок потрібних наборів даних у реальному часі, оскільки їх легше адаптувати до поточних подій.

Гарантія якості означає, що пошукові роботи краще оцінюють якість вмісту, тобто це інструмент, який надає перевагу, наприклад, під час виконання завдань із забезпечення якості.

Під час веб-сканування основним результатом зазвичай є списки URL-адрес. Можуть бути інші поля чи інформація, але зазвичай посилання є переважаючим побічним продуктом.

Що стосується вебскрапінгу, результатом можуть бути URL-адреси, але сфера набагато ширша й може включати різноманітні поля, наприклад:

- ціна товару/акції;
- кількість переглядів/уподобань/поширень (тобто соціальна залученість);
- відгуки покупців;
- зіркові рейтинги продуктів конкурентів;
- зображення, зібрані з галузевих рекламних кампаній;
- запити в пошуковій системі та результати пошукової системи в хронологічному порядку.

Незважаючи на різницю, пошукові роботи та вебскрапери мають спільні проблеми.

Блокування даних: багато веб-сайтів мають політику захисту від сканування/скрапінгу, що може ускладнити збір потрібних даних. Служба вебскрапінгу іноді може бути надзвичайно ефективною в цьому випадку, особливо якщо вона надає доступ до великих проксі-мереж, які можуть

допомогти у зборі даних за допомогою реальних IP-адрес користувачів і обходити ці типи блокувань.

Трудомісткість: виконання завдань сканування/скрапінгу даних у масштабі може бути дуже трудомістким і займати багато часу. Компанії, які, можливо, спочатку час від часу потребували наборів даних, але тепер потребують регулярного потоку даних, більше не можуть покладатися на ручний збір.

Обмеження збору даних: виконання сканування/скрапінгу даних зазвичай можна легко виконати для простих цільових сайтів, але коли користувач починає стикатися зі складнішими цільовими сайтами, деякі IP-блоки можуть бути нездоланними.

Знаючи різницю між пошуковими роботами і вебскраперами, користувач може обирати той вид видобутку веб-контенту, який найбільше відповідає конкретному випадку використання. Потрібно лише визначитися з необхідністю найму штатного персоналу, який буде керувати процесом збору даних, а також з бюджетом. В умовах обмеженості бюджету, а також в залежності від цілей, яких хочу досягти користувач під час видобутку даних, можна скористатися некомерційними програмними засобами веб-майнінгу. У наступних підрозділах буде проведено ознайомлення і порівняння популярних програмних засобів видобутку веб-контенту комерційної та некомерційної груп.

2.3 Комерційні програмні засоби видобутку веб-контенту

2.3.1 Easy Web Extract

Даний інструмент є потужним програмним продуктом із простими у використанні засобами для видобутку, вилучення інформації, і для веб-майнінгу. Однією з важливих особливостей цього інструменту є вилучення схожих даних, а також веб-даних, які містять шаблони вилучення [12]. Це

дозволяє користувачам, наприклад, створювати проекти для веб-сайтів із подібною структурою (такі як інтернет-магазини, сайти купівлі та продажу, пошукові системи тощо). Інтерфейс Easy Web Extract зображено на рисунку 2.1.

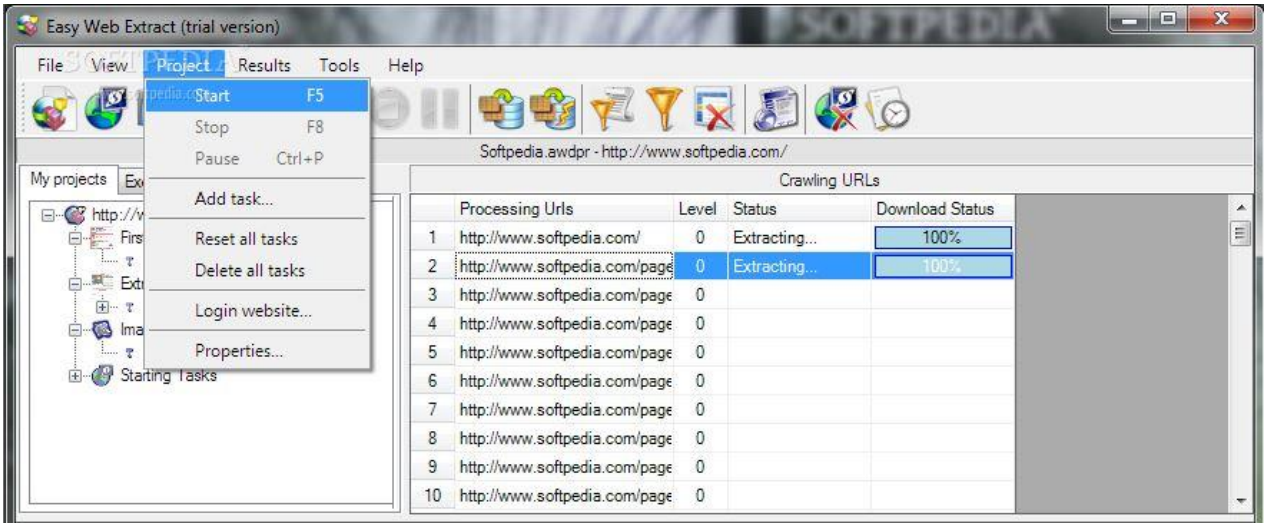


Рисунок 2.1 – Інтерфейс Easy Web Extract

Easy Web Extract має налаштований веб-сканер, в якому можна визначити правила сканування та мультиоб’єктного завантаження. Іншою важливою особливістю цього інструменту є доступ до інформації захищених паролем сайтів. Створення вилучених даних у різних форматах, таких як Excel (.csv), Text (.txt), HTML, файл XML, Microsoft Access, SQL, MySQL, є також однією із функцій цієї програми.

2.3.2 Web Content Extractor

Web Content Extractor — це програмне забезпечення для аналізу та вилучення даних. Цей інструмент може збирати дані з онлайн-магазинів, комерційних сайтів, бізнес-сайтів, сайтів купівлі та продажу, результатів пошукових систем тощо. Це програмне забезпечення дозволяє користувачеві представляти дані видобутку у формі Excel (.csv), тексту (ASCII), а також в HTML форматі та в форматі баз даних MySQL [13]. Деякі з переваг цього

інструменту налічують: використання веб-сканування і веб-павуків, скачування інформації як кількох суб'єктів, можливість збору даних із сайтів, захищених паролем, простоту використання, швидке і точне навчання тощо. Наприклад, на веб-сайтах пропозицій і фондових бірж цей інструмент може виявитись дуже корисним і важливим. На рисунку 2.2 зображено інтерфейс цієї програми.

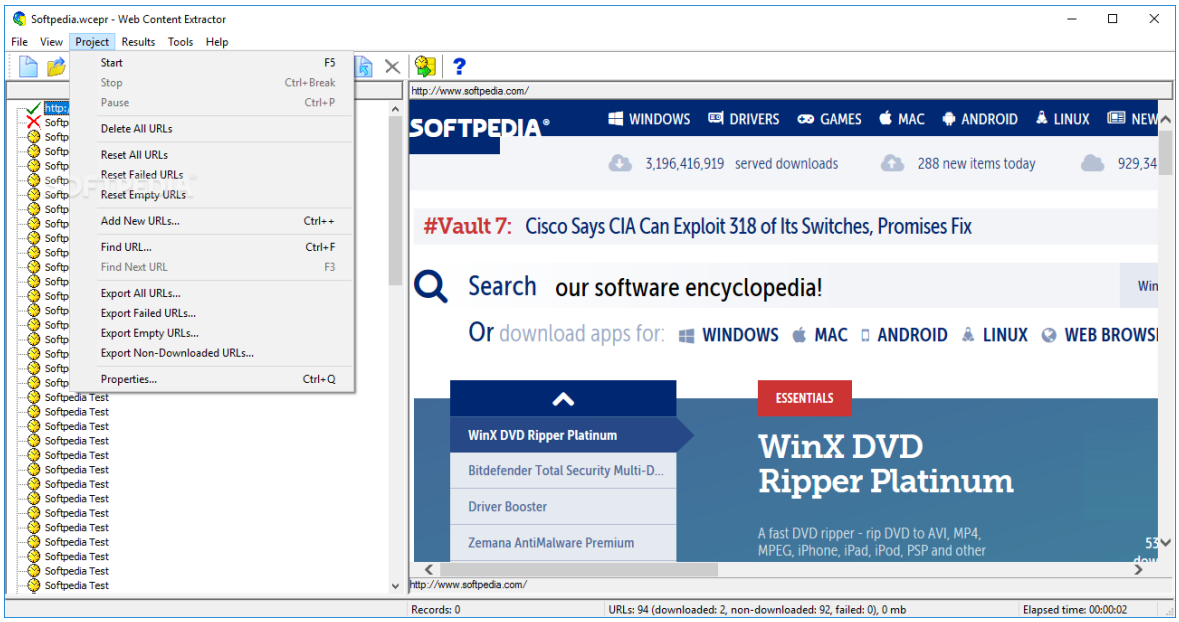


Рисунок 2.2 – Інтерфейс Web Content Extractor

Web Content Extractor має деякі переваги перед іншими програмами:

- допомагає в отриманні чи зборі ринкових цифр, даних про ціни на продукти або даних про нерухомість;

- допомагає користувачам отримувати інформацію про книги, включаючи їхні назви, авторів, описи, номери ISBN, зображення та ціни з онлайн-продавців книг;

- допомагає користувачам автоматизувати отримання аукціонної інформації з аукціонних сайтів;

- допомагає журналістам отримувати новини та статті з новинних сайтів;

- допомагає людям, які шукають оголошення про роботу на веб-сайтах з роботою в Інтернеті, знайти нову роботу швидше та з мінімальними

незручностями;

- допомагає в отриманні онлайн-інформації про відпустку та місця відпочинку, включаючи їх детальні описи з веб-сайтів.

2.3.3 Web Info Extractor

Web Info Extractor є потужним інструментом для видобутку даних і вилучення веб-контенту. Він витягує структуровані та неструктуровані дані з веб-сторінок і перетворює їх в локальний файл або зберігає у базі даних і надсилає все на веб-сервер. Цей інструмент має важливу функцію оновлення під час створення нового контенту [14]. Це проявляється в тому, що програма постійно відстежує веб-сторінку, і коли на сторінку додається новий контент, в залежності від змін, завдання, призначене інструменту, оновлюється. Вилучені бази даних можна зберегти у форматах .csv або .txt. На рисунку 2.3 зображено інтерфейс Web Info Extractor.

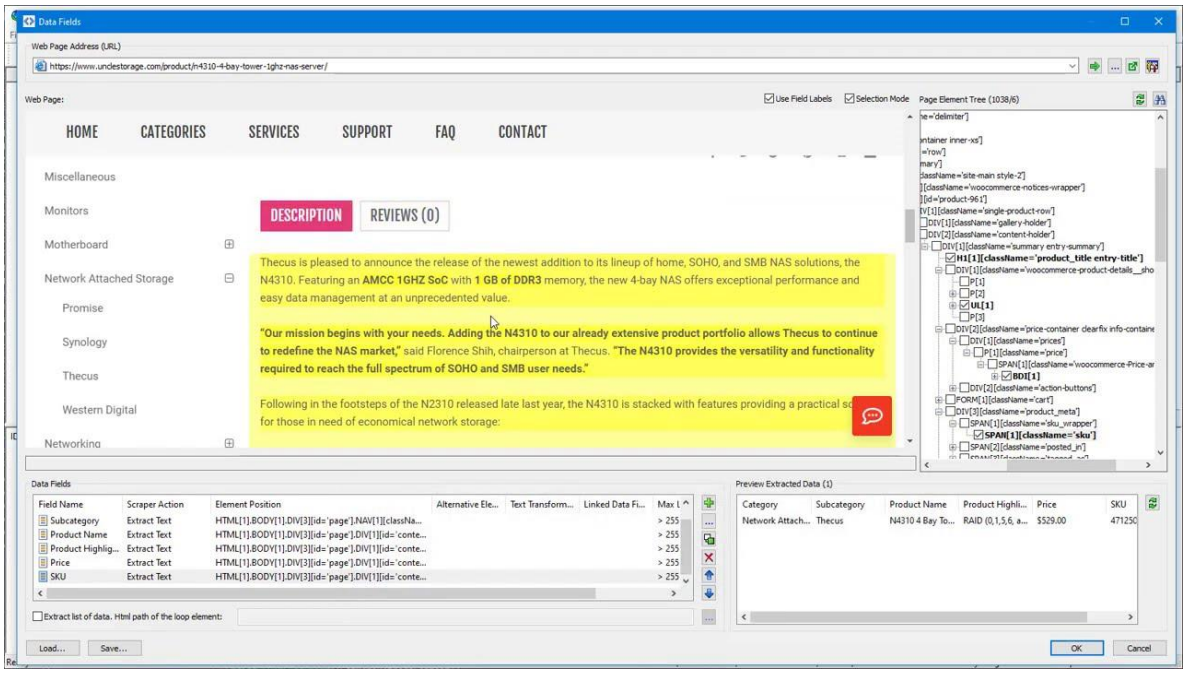


Рисунок 2.3 – Інтерфейс Web Info Extractor

У цьому інструменті залежно від типу інформації, якими є текст, посилання, зображення, вихідний код або список даних таблиці, етапи проекту та вибору параметрів відрізняються, але загалом кроки однакові. Наприклад, щоб вибрати зображення даних, спочатку виділяється зображення, а навколо нього створюється червона рамка. При виборі бажаних даних, вони відображаються в поточному об'єкті. Потім вибирається кнопка «створити елемент», і в щойно відкритому вікні користувач вводить назву, тип і атрибут елементів. Нарешті, за допомогою операції, наданої користувачем, вибрані дані на початкових етапах вилучаються та зберігаються в списку результатів. Недоліки цього інструменту полягають у довгому завантаженні веб-сайту та довгій тривалості роботи.

Переваги Web Info Extractor перед іншими інструментами:

- полегшує визначення інструментів вилучення, які дають змогу не вивчати нудні та складні правила шаблонів;
- полегшує рекурсивне визначення задачі;
- вилучення табличних і неструктурованих даних у файл або базу даних;
- вилучення нового контенту під час оновлення та моніторингу веб-сторінок;
- робота з текстом, зображеннями та іншими файлами посилання;
- робота з веб-сторінками всіма мовами;
- виконання багатозадачності в однаковий час.

2.3.4 Screen-Scraper

Screen-Scraper дозволяє комп’ютеру неодноразово отримувати символьні дані від центрального процесора та робити їх розпізнаваними для графічного інтерфейсу. Нові версії Screen-Scraper отримують дані з HTML, тож він може мати доступ до інформації за допомогою браузера.

Цей інструмент надає користувачам графічний інтерфейс, який дозволяє вказувати адресу елементів даних, які було видобуто, а також надає можливість

працювати з вилученими даними. Серед переваг цього програмного забезпечення є також здатність завантажувати всі продукти веб-сайту в електронну таблицю.

Screen-Scraper може легко збирати дані з веб-сторінок. У разі потреби отримання інформації з певного сайту, за допомогою Screen Scraper можна завантажити одночасно більшість файлів веб-сайту для доступу та порівняння [15]. Це програмне забезпечення забезпечує користувачам дуже легкий і зручний досвід використання, завдяки простому графічному інтерфейсу і розширеним інструментам вилучення, як показано на рисунку 2.4.

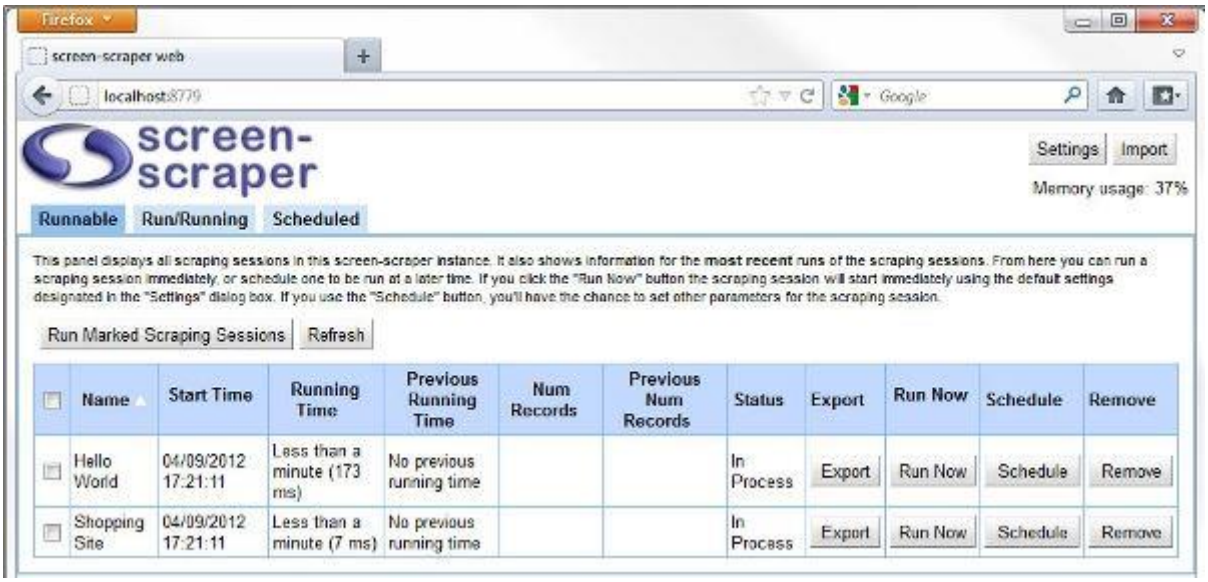


Рисунок 2.4 – Інтерфейс Screen-Scraper

Деякі з основних функцій інструмента Screen Scraper налічують: автоматичне копіювання тексту веб-сторінок, автоматичне відкриття посилань, автоматичне заповнення та надсилання форм на веб-сайтах, автоматичне завантаження файлів, таких як PDF, Word, зображення тощо з веб-сайтів, інтеграція або розділення інтегрованих даних більшістю мов програмування (Java, PHP, .NET, ASP), які можна запускати на сервері.

Переваги Screen-Scraper:

- Screen-scraper представляє собою графічний інтерфейс, який дозволяє

користувачеві виділяти URL-адреси, елементи даних, які потрібно вилучити, і логіку сценаріїв для перегляду сторінок і роботи з видобутими даними;

- після створення елементів із зовнішніх мов, таких як .NET, Java, PHP і ASP, можна викликати екранний очисник;
- спрощує збирання інформації з циклічними інтервалами, що допомагає отримувати дані про продукти та завантажувати їх в електронну таблицю;
- має в собі класифікатор, прикладом якого може бути метапошукова система, у якій пошуковий запит, введений користувачем, виконується одночасно на кількох веб-сайтах у режимі реального часу, після чого результати відображаються в єдиному інтерфейсі.

2.3.5 Web Data Extractor

З першого погляду можна побачити, що це простий у використанні та відносно комплексний інструмент. Він може видобувати різні дані, наприклад мета-теги, URL-адреси, електронну пошту, телефон, факс тощо. Стандартним форматом зберігання вилучених даних є .csv. Частина інтерфейсу цього інструменту представлена на рисунку 2.5.

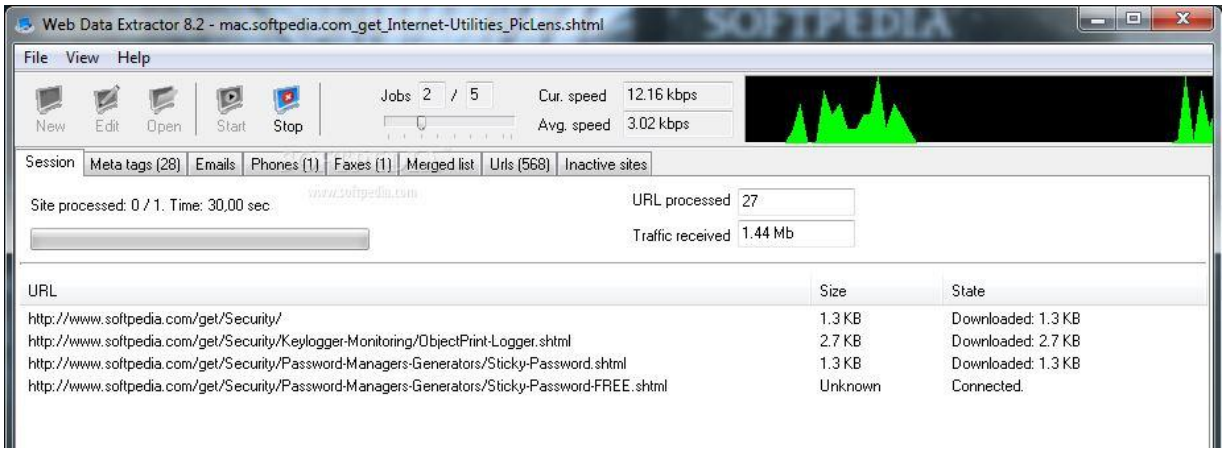


Рисунок 2.5 – Інтерфейс Web Data Extractor

Однією з важливих особливостей Web Data Extractor є можливість

змінювати параметри відповідно до вподобань користувача, а також обирати тип пошукової системи [14]. У цьому інструменті для редагування можна використовувати фільтри, які доступні в інструментах для вилучення URL-адреси, тексту, даних і парсерів. Наприклад, у фільтрі URL є частина, яка містить список акредитованих доменів для адрес веб-сайтів. Перевагою цього списку є можливість віднімання або додавання до/з нього. Щодо фільтру тексту та даних можливо, якщо потрібний текст і дані існують, веб-сторінку буде вилучено. У синтаксичному аналізаторі для телефону, факсу та електронної пошти розміщуються префіксні фрази, які зазвичай використовуються на веб-сайті для номерів телефонів, факсів та адрес електронної пошти.

Цей інструмент відрізняється від інших підходом до початку операції вилучення. Тут спочатку вибираються та встановлюються всі бажані налаштування, а потім дані відображаються відповідно до цієї конфігурації, тоді як в інших інструментах спочатку завантажується сайт, а потім користувач, залежно від вмісту та інформації, починає створювати свій шаблон і правило. Хоча в Web Data Extractor на початку роботи налаштування та доступні параметри є загальними, але для людини, ознайомленою із видобуванням за допомогою іншого програмного забезпечення, може бути корисним ознайомитися з різними параметрами безпосередньо на початку проекту. Ця функція може розширити погляд користувача на процедуру вилучення інформації в цілому.

2.3.6 Automation Anywhere

Працювати з цим програмним забезпеченням дуже легко, і користувач без будь-яких труднощів може легко вилучити дані залежно від потреби. Завдяки розумній та автоматизованій технології в цьому інструменті, інтерфейс якого зображено на рисунку 2.6, складні та трудомісткі операції виконуються швидко та не потребують програмування [16].

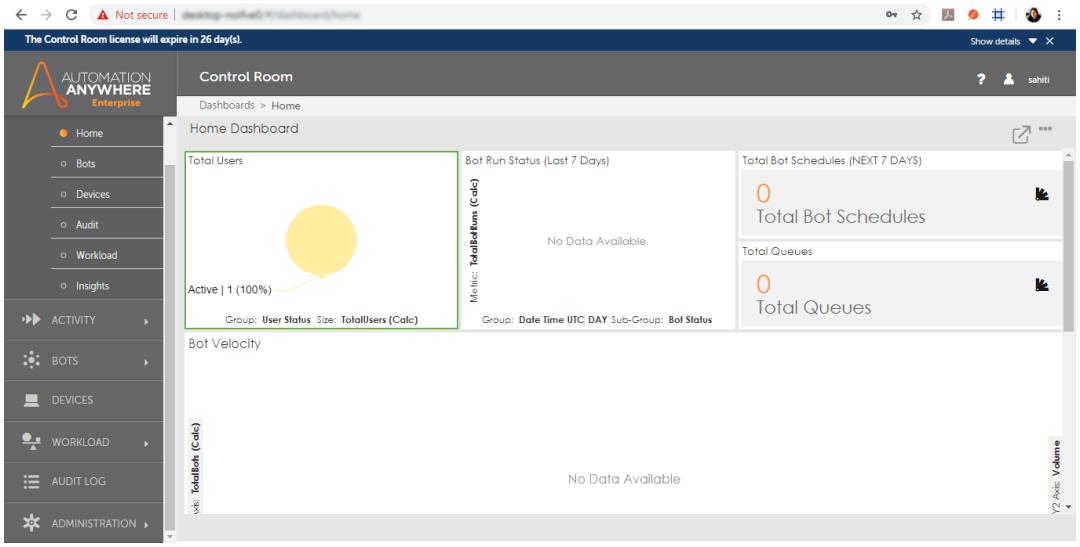


Рисунок 2.6 – Інтерфейс Automation Anywhere

Однією з особливостей цього інструменту є можливість повторювати дію протягом годин, хвилин або секунд. Також можна вказати швидкість необхідної дії. Також однією з важливих особливостей програми є наявність розділу під назвою «Планувальник». У цьому розділі можна запланувати необхідну дію, для виконання у вказаний час. Наприклад, можна налаштувати так, щоб вона виконувалась раз на день, в певні дні тижня або раз на місяць.

Крім запису дій у вікнах, цей інструмент може записувати в Інтернеті. Ця частина, яка називається Web Recorder, записує всі дії в Інтернеті від відкриття веб-сайту до вибору кнопки чи посилання. Використовуючи Automation Anywhere, можна витягувати два типи даних, включаючи численні дані та таблиці. Вихідні дані програми є у форматах xml, txt, excel, mysql.

Загалом веб-реєстратор цього інструменту можна використовувати для таких елементів:

- вхід на веб-сайт;
- заповнення та надсилання форм;
- оновлення записів бази даних;
- навігація необхідним пошуком;
- використання веб-базового erp;

- вилучення даних з Інтернету;
- тестування онлайн-програм.

2.3.7 Mozenda

Mozenda − популярний інструмент із високим рейтингом. Це програмне забезпечення, яке дозволяє професійним і непрофесійним користувачам легко отримувати дані з веб-сторінок. Залежно від можливості вибору веб-контенту та необхідності використання хмарних обчислень, у цьому інструменті вилучення, зберігання та керування даними здійснюються у централізованій та комплексній формі.

Однією з особливостей цього програмного продукту є автоматичне вилучення даних без залишення будь-яких ознак [17]. Насправді, це відбувається через використання Mozenda невідомого проксі-сервера, який може створювати змінну IP-адресу та запобігати ідентифікації користувача. Крім того, інструмент Mozenda, як і інструмент Automation Anywhere, дозволяє користувачеві налаштовувати час вилучення даних. Інтерфейс Mozenda показано на рисунку 2.7.

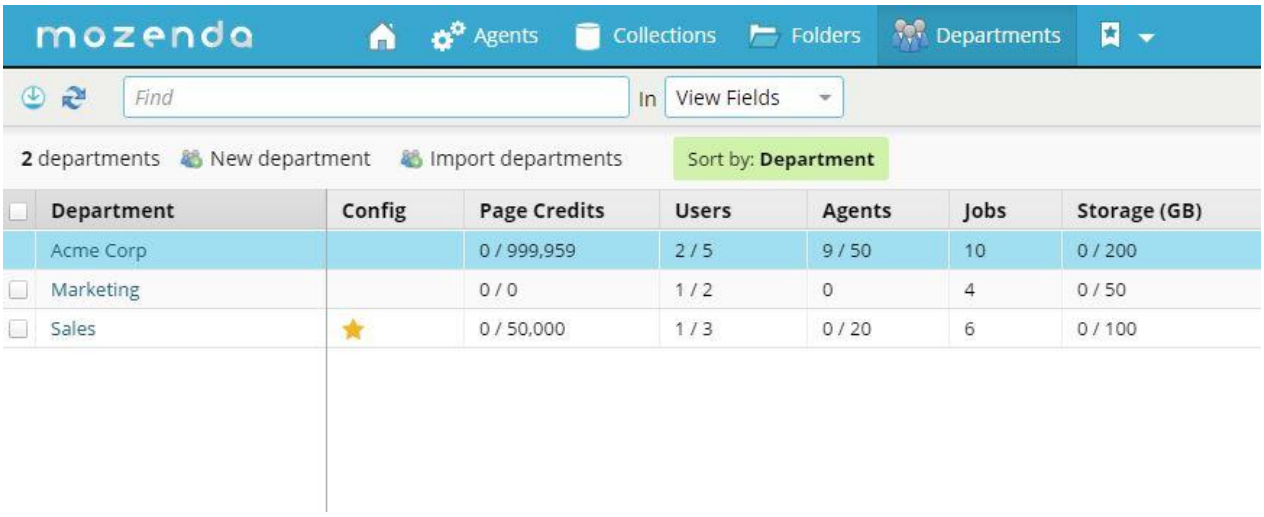


Рисунок 2.7 – Інтерфейс Mozenda

Ті частини цього інструменту, які пов’язані з майнінгом даних, включають повний пакет корисних програм для маркетологів. Використовуючи функції Mozenda, користувач може робити всі справи, пов’язані з прогнозами, отримувати інформацію для створення бюджету, дослідження, аналізу та ціноутворення конкурента тощо.

Інструмент Mozenda за допомогою текстового фільтру здатний швидко фільтрувати текст користувача та вилучати певні його частини. Як зазначалося раніше, ця функція подібна до характеристик інструменту Web Data Extractor. У Mozenda є два розділи:

Розділ веб-консолі: ця програма дозволяє користувачеві запускати агентів, упорядковувати та публікувати результати даних, отриманих із веб-середовища.

Розділ Agent Producer: це програма для Windows, яку можна використовувати для створення проекту, пов’язаного з вилученням даних у Windows.

Перевагами Mozenda перед іншими інструментами є:

- простота у використанні;
- незалежність від платформи (працює лише на Windows);
- незалежність від робочого місця: налаштування скрапера, керування процесом збирання та отримання зібраних даних з будь-якого комп’ютера, підключеного до Інтернету.

2.4 Некомерційні програмні засоби видобутку веб-контенту

2.4.1 Import.Io

Хоча цей інструмент є однією із некомерційних програм у сфері веб-майнінгу, але її графічне середовище, як показано на рисунку 2.8, набагато краще, ніж у комерційних інструментів. В ньому є функція, яка дає змогу користувачеві вибрати тип вилучених даних щодо одного з параметрів

конектора, екстрактора та сканера [14]. Хоча продуктивність цих трьох варіантів однакова, вони мають деякі відмінності. Якщо дані та інформація мають шаблон після вилучення даних сторінки у формі таблиці, параметр «екстрактор» визначає алгоритми інтелектуального аналізу даних цього відношення та без будь-якого втручання користувача витягує інші пов'язані дані з шаблоном у сторінку та розміщує їх у таблиці.

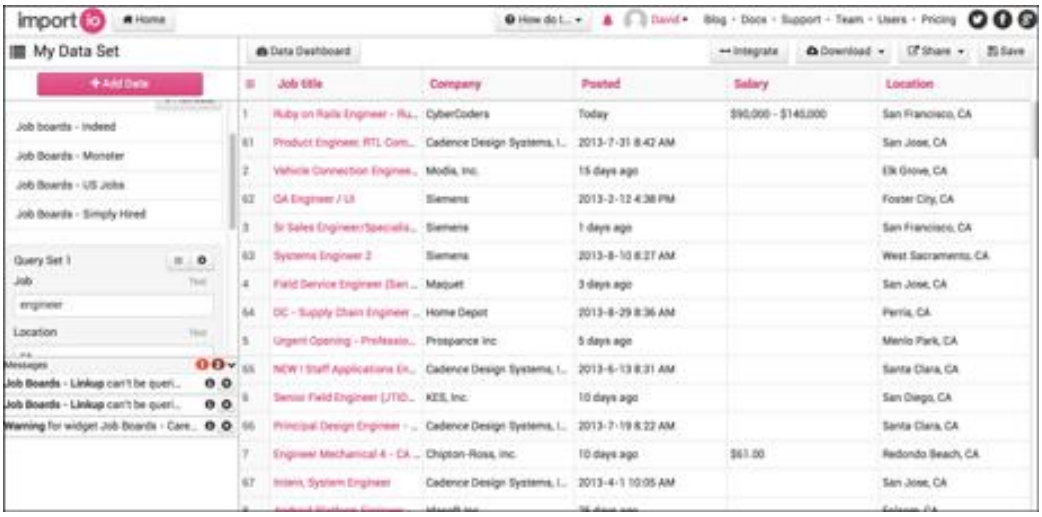


Рисунок 2.8 – Інтерфейс Import.Io

Щодо сканера, існує така функція, що, розробляючи шаблон для вилучення однієї з веб-сторінок, користувач також може виконувати вилучення даних на інших веб-сторінках. У опції Connector кожен користувач може отримати доступ до структурованих даних із отриманих результатів пошуку. Після пошуку Import.io зберігає дані в таблиці в структурованому вигляді. З цих таблиць можна отримати бажані дані, і за потреби, пізніше, використовуючи збережені дані, веб-сайт буде досліджено знову.

Цей інструмент витягує структуровані дані, де вони потім зберігаються на віртуальних серверах. Кожного разу, коли інформація розміщується на платформі цього інструменту, створюється API для доступу до секретних даних, тому онлайн-доступ до інтегрованих даних може бути здійсненим з легкістю в будь-який час.

2.4.2 Irobotsoft

Цей інструмент є портативним, тому його не потрібно встановлювати для використання. Він представляє собою інтелектуального робота для виконання всіх дій, пов’язаних із веб-сайтом, таких як заповнення форм для членства, вибір посилань і підключення до бази даних. Для використання цього інструменту програмування не потрібне, і кожен без будь-яких знань в цій області може навчитися його використовувати. Але якщо користувач володіє хоча б невеликими навичками програмування, він зможе створити більш потужний інструмент irobot.

Irobotsoft працює на різних операційних системах, таких як XP, 7, Vista, Nt, і для його роботи потрібен браузер Internet Explorer. Однією з особливостей цього інструменту є багаторазове автоматичне вилучення даних з різних веб-сайтів [14]. Витягнуті дані можна зберігати у форматі csv і xml. Частину інтерфейсу Irobotsoft можна побачити на рисунку 2.9.

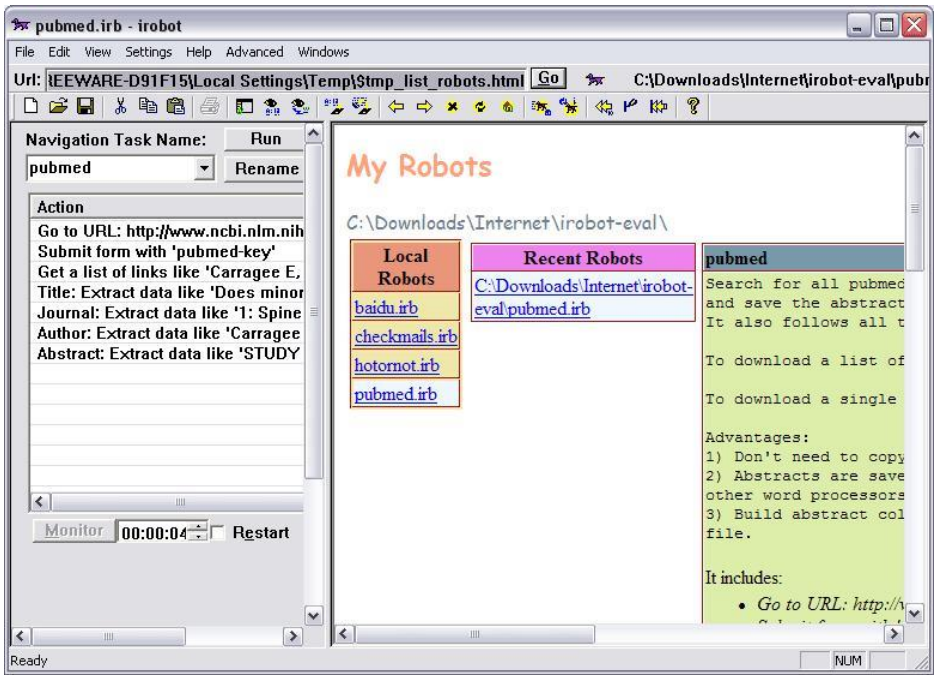


Рисунок 2.9 – Інтерфейс Irobotsoft

2.4.3 Webextractor360

Цей інструмент є повністю безкоштовним і має дуже просте середовище, як показано на рисунку 2.10. Після введення адреси потрібного сайту деякі параметри, такі як зображення, слова, таблиці, інтернет-адреси, електронна пошта, телефон, факс тощо, будуть відображені в інструменті, який може отримувати різноманітну інформацію з веб-сайту. Цей інструмент містить опції для вилучення, які в деяких випадках можна порівняти з комерційними програмами, такими як Web Data Extractor [14]. Наприклад, у Webextractor360 можна ігнорувати певні гіперпосилання, визначені для пошуку, а також розміщення адрес сторінок у результатах пошуку є необов'язковим.

Хоча Web Data Extractor має більше можливостей і гнучкості, проте можна порівняти два наведені випадки для Webextractor360 із функцією фільтрації в Web Data Extractor.

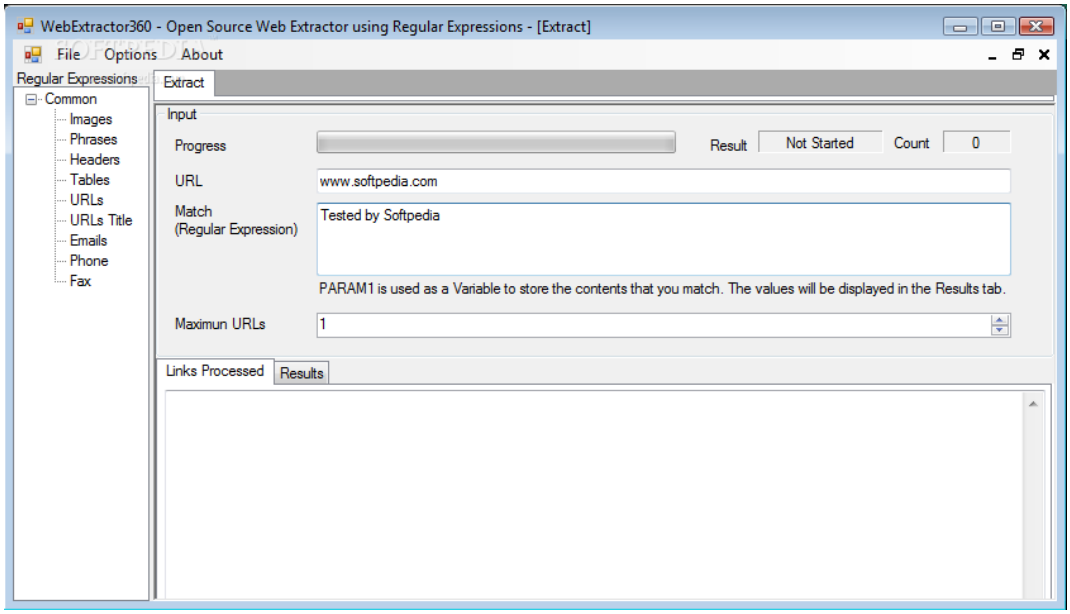


Рисунок 2.10 – Інтерфейс Webextractor360

2.4.4 Scrapy

Scrapy — це безкоштовний інструмент для вилучення веб-даних, який витягує структуровані дані з веб-сторінок. Він підходить для різних цілей, наприклад для автоматичного тестування веб-сторінок, моніторингу та аналізу даних [14]. Scrapy написаний на мові Python і є портативно виконуваним в операційних системах Linux, Windows, Mac і BSD. Перед запуском Scrapy необхідно встановити Python в операційну систему і внести в неї деякі зміни (в залежності від типу системи). Однак використовувати цей інструмент трохи складніше, ніж інші.

2.4.5 Context Miner

Цей інструмент безкоштовно видобуває веб-контент онлайн. Вихідними форматами видобутих даних є xml і csv. Відмінність цієї програми від інших полягає в веб-майнінгу певних сайтів і ресурсів. Іншими словами, цей інструмент видобуває дані лише з таких сайтів і платформ, як, наприклад, Twitter, Youtube і Flickr.

2.5 Порівняння програмних засобів видобутку веб-контенту

Основною метою даного підрозділу є порівняння інструментів видобутку веб-контенту комерційної та некомерційної груп. Ціллю такого розділення є бажання показати, що в залежності від мети, яку ставить користувач, видобуваючи контент, некомерційні програмні засоби не поступаються в функціоналі комерційним засобам веб-майнінгу.

Таблиця 2.1 демонструє результати порівняння інструментів видобутку веб-контенту по критеріям операційної системи, комерційності, типу даних для видобутку і формату даних експорту.

Таблиця 2.1 – Порівняння інструментів видобутку веб-контенту по різним критеріям

Назва програми	Комерційність	Операційна система	Дані для видобутку	Формат експорту даних
Easy Web Extract	Так	Windows	Текст, зображення, url, html	Microsoft Excel, Access, txt, odbc, sql, mysql, html, xml
Web Content Extractor	Так	Windows	Текст, зображення, мультимедіа	
Web Information Extractor	Так	Windows 2000/XP/2003	Структуровані або неструктуровані дані	txt, csv
Screen-Scraper	Так	Windows	Текст, зображення	
Web Data Extractor	Так	Windows 95/98/NT/2000/Me/XP	Мета теги, текст, url, телефони, ел. адреси, факс	Microsoft Excel
Automation Anywhere	Так	Windows	Неструктуровані дані	txt, xml, excel, mysql
Mozenda	Так	Windows	Текст, зображення, ціни, дата, адреси, телефони, факс	csv, tsv, xml
Import.Io	Ні	Windows,osx,linux	Текст, номери, локації, зображення, url	csv, html, xls
Irobotsoft	Ні	Windows XP/7/ Vista/Nt	Структуровані або неструктуровані дані	csv, xml
Webextractor360	Ні	Windows 2000/XP/2003/Vista	Зображення, фрази, html заголовки, html таблиці, url, телефони, ел. адреси, факс	txt
Scrapy	Ні	Windows, linux , mac, BSD	Структуровані дані	csv, xml,JSON(JavaScript Object Notation)
Context Miner	Ні	-		csv, xml

2.6 Мова Python як інструмент видобутку веб-контенту

Існують задачі, які важко виконати, використовуючи лише стандартні програмні засоби видобутку веб-контенту. Це можуть бути задачі, націлені на конкретну специфічну задачу, важкі для виконання з точки зору логіки програми, але зрозумілі для людини. Тут в нагоді стають мови програмування, де можна побудувати власні складні алгоритми вирішення завдань, в яких можна маніпулювати вхідними та вихідними даними як завгодно. Однією з таких мов є мова Python [18].

Python вважається найпоширенішою мовою програмування для виконання задач в галузі data mining. Це можна пояснити тим, що Python постачається з величезною групою бібліотек [19, 20], включаючи модулі для машинного навчання.

Що робить цю мову програмування найкращим вибором для веб-видобутку, так це її здатність обробляти практично всі процеси, пов’язані з вилученням даних. Крім того, що Python простий у використанні (тут, наприклад, не використовуються крапки з комою та фігурні дужки), він також відрізняється прямим використанням змінних, де це необхідно, що значно полегшує та прискорює роботу. Мова програмування також відома своїм підходом «маленький код, велике завдання», згідно з яким коди, як правило, малі порівняно з кодами інших програм.

Крім того, синтаксис даної мови дуже простий для розуміння. Її читання схоже на читання англійських фраз і тверджень. Програмісти-початківці та навіть ті, хто нічого не знає про програмування на Python, швидше за все зрозуміють або матимуть уявлення про те, для чого призначена та чи інша частина коду.

Також допомагає те, що Python має величезну глобальну спільноту користувачів. Існує багато дискусійних дошок і чат-груп, присвячених програмуванню на цій мові, де користувачі можуть легко звернутися за допомогою чи порадою щодо того, як впоратися з труднощами, з якими вони

стикаються під час написання своїх програм та алгоритмів з видобутку контенту веб-сайтів.

В даному підрозділі буде проведено огляд та порівняння основних бібліотек та інструментів Python 3 для видобутку веб-контенту.

2.6.1 Бібліотека BeautifulSoup

BeautifulSoup − це найпопулярніша бібліотека Python, яка допомагає розбирати документи HTML або XML у деревовидну структуру, щоб знаходити та витягувати дані з веб-сторінок. Вона витягує інформацію у вигляді дерева, а пізніше допомагає використовувати дані у формі словників. Цей інструмент має зрозумілий інтерфейс взаємодії і автоматичне перетворення кодування, щоб полегшити роботу з даними веб-сайту. Він дуже простий в освоєнні та опануванні, а також має хорошу вичерпну документацію, яка допомагає в його легкому вивченні.

Програмісти на Python, які використовують BeautifulSoup, можуть отримувати вихідний код веб-сторінки та відфільтровувати його для знаходження потрібної інформації. Наприклад, цей інструмент може виявляти елементи HTML за ідентифікатором або назвою класу та виводити знайдені дані для подальшої обробки чи переформатування. Фільтрування сторінки за допомогою селекторів CSS є корисною стратегією копіювання, яку відкриває ця бібліотека.

BeautifulSoup − це бібліотека синтаксичного аналізу, яка також добре справляється з отриманням вмісту з URL-адреси та дозволяє аналізувати певні його частини без жодних проблем, але при цьому має певні недоліки. Вона лише отримує вміст URL-адреси, наданої користувачем, і зупиняється на цьому, тобто не буде проводити сканування далі, тільки якщо вручну не помістити код в нескінченний цикл із певними критеріями.

Для повноцінної роботи BeautifulSoup потрібні інші залежності Python. Наприклад, користувачеві знадобиться бібліотека запитів, щоб отримати

вихідний код HTML-сторінки у власний сценарій, перш ніж користувач зможе почати його аналіз. BeautifulSoup дуже простий у запуску та відносно простий у використанні. Він ідеально підходить для невеликих проектів, де уже відома структура веб-сторінок для аналізу.

Нижче коротко перелічені основні переваги BeautifulSoup [19, 20]:

- легкий у вивченні та опануванні початківцями галузі веб-скрапінгу;
- має хорошу підтримку спільноти для знаходження проблем;
- має хорошу вичерпну документацію.

Даний модуль не є вбудованим у Python. Щоб встановити його, потрібно ввести наведену нижче команду в терміналі:

```
pip install BeautifulSoup4.
```

2.6.2 Бібліотека Scrapy

Scrapy − одна з найпотужніших бібліотек Python. Це спільний фреймворк з відкритим вихідним кодом для отримання даних із потрібних веб-сайтів. Scrapy має швидку продуктивність і надає вбудовану підтримку для отримання даних із джерел HTML або XML за допомогою виразів CSS і виразів XPath [19].

Scrapy – це фактично повний фреймворк веб-скрапера. В ньому користувач може надавати Scrapy кореневу URL-адресу для ініціалізації операції копіювання, щоб потім (за бажанням) вказати, скільки URL-адрес він бажає сканувати та отримати.

Нижче коротко перелічені основні переваги Scrapy [19]:

- він легко розширюється;
- має вбудовану підтримку вилучення даних;
- має дуже високу швидкість, порівняно з іншими бібліотеками;
- ефективний у використанні пам'яті та ЦП;
- використовується для створення надійних та розширених програм;
- має сильну підтримку громади.

Якщо користувач має справу зі складною операцією скрапінгу, яка

потребує величезної швидкості та складності, то йому слід віддати перевагу Scrapy, а якщо мова йде про новачка у програмуванні, який хоче працювати з проектами веб-скрапінгу, тоді BeautifulSoup підійде краще, оскільки в ньому можна легко навчитися виконувати операції з високою швидкістю.

Даний модуль не є вбудованим у Python. Щоб встановити його, потрібно ввести наведену нижче команду в терміналі:

```
pip install scrapy.
```

2.6.3 Бібліотека Selenium

Selenium − це універсальний інструмент візуалізації веб-сторінок, призначений для автоматизованого тестування. Selenium слід розглядати як веб-браузер, який виконує JavaScript і повертає HTML у сценарій користувача. Його широка підтримка популярних мов програмування означає, що програмісти можуть самі обрати ту мову, яка їм подобається найбільше. Користувачі Python можуть імпортувати веб-драйвер Selenium, щоб розпочати автоматичне сканування через різноманітні локатори, такі як: ID, name, className, tagName, linkText, partialLinkText, CSS selector і XPath [20].

Нижче коротко перелічені основні переваги Selenium [20]:

- працює з динамічним контентом;
- підтримує сценарії;
- більш гнучкий інструмент, ніж BeautifulSoup;
- підтримка декількох операційний систем (Windows, Linux, MacOS);
- підтримка декількох мов програмування (Java, C#, Ruby та ін.);
- підтримка декількох браузерів.

Selenium є чудовим варіантом скрапінгу у випадках, коли сторінку потрібно завантажити заздалегідь, перш ніж JavaScript зможе відобразити динамічний вміст. Це гнучкий інструмент для особливих випадків, в яких необхідна система автоматизації, що може виконувати такі дії, як натискання кнопок і вибір спадних меню. Він досить універсальний, щоб працювати в

кількох браузерах, операційних системах і навіть апаратних пристроях, таких як телефони Blackberry та Android. Така гнучкість є головною перевагою Selenium разом із природою проекту з відкритим кодом, що заохочує розробку плагінів.

Розробникам слід пам’ятати про деякі недоліки використання Selenium для своїх проектів веб-скрапінгу. Найпомітнішим недоліком є те, що він не такий швидкий, як HTTPS-запити BeautifulSoup. Усі веб-сторінки мають спочатку завантажитися, перш ніж Selenium почне діяти, і кожна команда Selenium має спочатку пройти через протокол JSON HTTP. Використання пропускної здатності є високим через завантаження повних веб-сторінок, як і використання ЦП через повторне виконання JavaScript. Також веб-скрепери повинні знати, що сценарії Selenium часто можуть ламатися через поверхневі зміни інтерфейсу.

Для встановлення даного модулю в Python, потрібно ввести наведену нижче команду в терміналі:

```
pip install selenium.
```

2.6.4 Порівняння бібліотек Python

Кожна з наведених вище бібліотек Python має свої переваги та недоліки, і вибір між використанням однієї з них залежить від суті, масштабу та цілей проекту з видобутку контенту, а також від ресурсів системи, на якій працює користувач.

Так, наприклад, якщо мова йде про простий проект, тоді BeautifulSoup є найкращим вибором, але якщо користувач виконує складну операцію скрапінгу, то йому слід віддати перевагу Scrapy або Selenium. Тобто, вибір між використанням цих технологій видобутку веб-контенту, ймовірно, відображатиме обсяг проекту [19, 20].

Коли справа доходить до вибору лише однієї бібліотеки, Selenium або Scrapy, рішення в кінцевому підсумку зводиться до природи варіантів

використання. Кожна бібліотека має свої плюси і мінуси. Selenium − це насамперед інструмент веб-автоматизації, однак Selenium WebDrivers також можна використовувати для збирання даних із веб-сайтів, якщо ці сайти використовують JavaScript. З іншого боку, Scrapy — це потужний фреймворк для збирання даних, який можна використовувати для збирання величезних обсягів інформації із різних веб-сайтів.

Selenium вилучає відносно складніші динамічні сторінки за більшу вартість обчислювальних ресурсів. Проте почати роботу з BeautifulSoup легше, і, незважаючи на те, що кількість веб-сайтів, які він може отримати, є більш обмеженою, він ідеальний для проектів, де вихідні сторінки добре структуровані.

Стосовно швидкості операцій скрапінгу, Selenium очікує, поки клієнтські технології, такі як JavaScript, завантажаться в першу чергу, по суті, чекаючи завантаження повної сторінки. BeautifulSoup просто сканує джерело сторінки, що забезпечує швидше копіювання. Scrapy ж є асинхронним. Це означає, що він виконує декілька запитів одночасно. Навіть, якщо запит завершується невдачею або виникають будь-які помилки, це не впливає на вхідні запити. Це покращує загальну швидкість процесу. Selenium також надійний, але у випадку великого обсягу даних загальний процес відбувається повільно.

Selenium працює як безголовий браузер [20]. Він може функціонувати як комплексний набір інструментів веб-автоматизації, який імітує клацання мишею та заповнення форм. Уся ця потужність означає, що розробники мають крутішу криву навчання. BeautifulSoup може змістовно взаємодіяти лише з менш складними сторінками, але ним простіше користуватися: користувач може почати роботу з сайтами за допомогою BeautifulSoup лише за допомогою кількох рядків коду.

Selenium підтримує взаємодію з динамічними сторінками та контентом. Це є одночасно і перевагою, і недоліком. Він може працювати в більш широкому діапазоні сценаріїв, але поверхневі зміни інтерфейсу веб-сайту можуть зірвати сценарії, з якими може працювати BeautifulSoup. Хоча

BeautifulSoup по суті обмежується вилученням даних зі статичних сторінок, але простота інколи є перевагою, оскільки він більш стійкий до змін зовнішнього дизайну, оскільки переглядає лише вихідний код сторінки.

Selenium досить гнучкий, щоб робити майже все, що може BeautifulSoup [20]. Наприклад, він може знаходити багато тих самих структурованих елементів, що і BeautifulSoup, за допомогою команди driver.find_element_by_xpath. Незважаючи на те, що Selenium є більш гнучким, все одно вважається найкращою практикою використовувати його лише там, де це необхідно, щоб обмежити використання ресурсів.

Інший приклад, користувач працює над проектом, де потрібно вилучити великі обсяги даних з різних веб-сайтів. Щоб перевірити ці веб-сайти, потрібно зробити кілька викликів за допомогою проксі та VPN. Крім того, потрібен надійний механізм без затримок. У таких ситуаціях Scrapy є ідеальним вибором. За допомогою Scrapy можна легко працювати з проксі та VPN. Він може отримувати великі обсяги даних, оскільки це спеціалізований фреймворк для збирання веб-сайтів.

Щоб допомогти собі візуалізувати власну стратегію копіювання, може бути корисним скористатися опцією меню інструментів розробника свого браузера. Це допоможе побачити структуру сайту для скрапінгу завдяки демонстрації об’єктної моделі документа (DOM) веб-сайту. Навігація DOM дозволить обирати об’єкти HTML і XPath для націлювання.

Наприклад, гіпотетичною метою скрапінгу користувача є веб-сторінка, яка завантажує динамічний вміст. Крім того, він хоче взаємодіяти з веб-сторінкою перед її копіюванням. Незважаючи на те, що динамічний вміст з автоматизованою взаємодією знаходиться прямо всередині функцій Selenium, користувач хоче використовувати його лише для того, щоб веб-сторінка відображала його джерело. В такому разі передача Selenium фактичного аналізу BeautifulSoup після завантаження потрібної сторінки та розкриття DOM дозволяє обмежити використання ресурсів.

В таблиці 2.2 наведено короткі результати порівняння бібліотек

BeautifulSoup, Scrapy і Selenium [19, 20] за різними критеріями.

Таблиця 2.2 – Порівняння бібліотек BeautifulSoup, Scrapy і Selenium

Критерій	BeautifulSoup	Scrapy	Selenium
Структура	Бібліотека	Повноцінний фреймворк	Інструмент рендерінгу сторінки
Продуктивність	Деякі завдання виконує досить повільно	Має вбудовані функції, завдяки яким може виконувати роботу досить швидко	Працює повільно через необхідність повної загрузки сторінки
Розширюваність	Найкраще підходить для невеликих проектів	Краще підходить для великих та складних проектів	Краще підходить для великих та складних проектів
Дружелюбність до початківців	Найкращий вибір для початківців	Scrapy є більш складним у порівнянні з BeautifulSoup	Selenium є більш складним у порівнянні з BeautifulSoup
Спільнота	Спільнота розробників відносно слабка	Спільнота розробників Scrapy сильна та обширна	Середня спільнота розробників
Огляд	Розглядається як парсер	Вважається веб-сканером	Зручний інструмент для роботи зі скриптами

Підводячи підсумок, можна сказати, що BeautifulSoup, Scrapy і Selenium є гарними варіантами для веб-видобутку, але проблемою кожного веб-скрапара є

мінливість, притаманна мережі. Стратегія копіювання, яка працює на одному сайті, може не працювати на іншому. І самі веб-сайти можуть змінюватися, через що сценарії можуть виводити потім помилки під час наступних запусків. Саме тому автономні пошукові роботи потребують регулярного обслуговування.

2.7 Показники ефективності веб-видобутку

Для оцінки ефективності видобутку веб-контенту часто використовують різні показники продуктивності. Першим і найголовнішим із таких показників є пропускна здатність. Вона визначається як загальний час, необхідний для виконання даних веб-контенту.

Є і інші показники ефективності видобутку веб-контенту [3], перелічені нижче.

Ефективність витрат: вимірюється як відношення пропускної здатності до вартості видобутку веб-контенту.

$$\text{Ефективність витрат} = \frac{\text{Пропускна здатність}}{\text{Витрати видобутку}} \qquad (2.1)$$

Масштабування: це здатність системи керувати більшою кількістю даних видобутку веб-контенту з інтеграцією більшої кількості комп’ютерів, зберігаючи при цьому продуктивність.

$$\text{Масштабування} = \frac{\text{Пропускна здатність після}}{\text{Пропускна здатність до}} \qquad (2.2)$$

Затримка: це час, необхідний для виконання набору операцій даних видобутку веб-контенту.

$$\text{Затримка} = \frac{1}{\text{Пропускна здатність}} \tag{2.3}$$

Довговічність: це здатність системи зберігати інформацію протягом тривалого періоду часу.

$$\text{Довговічність} = \frac{\text{Поточні зчитки}}{\text{Загальна кількість зчиток}} \tag{2.4}$$

Одночасність: це здатність системи надавати послуги різним користувачам одночасно.

$$\text{Одночасність} = \frac{\text{Успішні операції}}{\text{Загальні операції}} \tag{2.5}$$

Наведені вище показники є лише декількома із багатьох способів вимірювання продуктивності видобутку веб-контенту та виконання операцій з даними [3].

2.8 Висновки

У даному розділі було представлено концепцію інструментів видобутку веб-контенту та необхідність їх використання для удосконалення пошуку та знаходження корисної інформації в Інтернеті. Для вилучення необхідних даних, користувачам доступна велика кількість інструментів видобутку. У дослідженні було представлено неповний список доступних інструментів, якими користуються професіонали та аматори у сфері веб-майнінгу, а також встановлено деякі об’єктивні критерії для їх порівняння.

Інструменти, розроблені для видобутку веб-контенту, можна умовно розділити на дві категорії: комерційні та некомерційні. Критерієм розділення

саме на такі категорії було бажання показати, що в залежності від цілей, які ставить користувач під час видобутку даних, некомерційні програмні засоби не поступаються в функціоналі комерційним засобам веб-майнінгу. Однією з головних задач даного розділу було ознайомлення та порівняння популярних програмних продуктів з цих двох груп. Результати порівняння програм за декількома критеріями були занесені в таблицю.

Через мінливість веб-сайтів, а також даних, розміщених на них, існують задачі, які важко виконати, використовуючи лише стандартні програмні засоби видобутку веб-контенту. Бібліотеки BeautifulSoup, Scrapy і Selenium мови програмування Python можуть виступати в якості доповнення або заміни програм з автоматичного видобутку веб-вмісту, завдяки набору функцій керування даними, доступному для користувача. Результати порівняння цих бібліотек за різними критеріями занесено до таблиці.

Оцінку ефективності видобутку веб-контенту можна провести за різними показниками, такими як: пропускна здатність, ефективність витрат, масштабування, затримка, довговічність і одночасність тощо.

Як висновок, є підстави вважати, що дослідження в галузі веб-майнінгу є багатообіцяючими, а також складними, і ця сфера допоможе створювати програми, які зможуть ефективніше використовувати мережу знань.

3 ВИРІШЕННЯ ПРОБЛЕМ АВТОМАТИЧНОГО ВИДОБУТКУ ВЕБ-КОНТЕНТУ

Процес видобутку контенту веб-сторінок є частиною галузі data mining і є необхідним етапом попередньої обробки, від якого значною мірою залежить якість отриманих знань. Навіть така проста задача, як вилучення тексту з контенту з веб-сторінки пов'язана з певною складністю – для цього потрібно попередньо відсікти непотрібні деталі першоджерела, такі як: «шапка» сайту, верхнє і вертикальне меню, рекламні блоки, блоки навігації, нижня частина сайту із зазначенням правовласника, студії-розробника, контактів, нижнє меню та інші елементи дизайну, що не містять, власне, корисної інформації. Це лише один з прикладів, але насправді проблем значно більше, ніж може здатися на перший погляд.

Даний розділ присвячений опису частих проблем і деяких їх рішень при автоматизованому вилученні даних для їх подальшого глибинного аналізу з мережі Інтернет.

3.1 Проблеми автоматичного видобутку веб-контенту та їх рішення

Усі методи вилучення даних із сайтів можна розбити на три основні групи: ручні, напівавтоматичні та автоматичні [21].

Ручні методи – це методи, в яких моніторинг потрібних сторінок і переміщення необхідної інформації в базу здійснюються спеціальним оператором (людиною). Перевагами такого підходу є висока якість видобутку інформації і низькі вимоги до кваліфікації оператора. Серед недоліків – висока трудомісткість, загальна незначна швидкість роботи і, найголовніше, людський фактор. При такому методі, в завданнях, що потребують високої точності вилучення та систематизації даних, кількість помилок може перевищити допустимий поріг, що є неприйнятним.

До групи напівавтоматичних методів належать рішення, за допомогою яких можна отримувати інформацію після певних налаштувань. Одними з найбільш поширених універсальних комерційних програмних засобів для напівавтоматичного вилучення даних є уже розглянуті в попередньому розділі Web Info Extractor, Web Data Extractor, Mozenda та інші. Також сюди можна віднести метод попередньої розмітки оператором елементів сторінки в графічному інтерфейсі користувача, і більш складні випадки – наприклад, складання регулярних виразів або запитів на мові XQuery. До переваг подібної групи методів слід віднести, в ідеалі, таку ж високу якість вилучення, як і при ручному способі, але набагато більш високу швидкість роботи. Серед недоліків можна виділити вже згаданий людський фактор, але у більш м'якій формі. Створена оператором розмітка може бути помилковою, тому потрібна початкова і періодична участь оператора, при чому його кваліфікація повинна відповідати певним вимогам (наприклад, йому слід вміти складати складні регулярні вирази).

У автоматичних (інтелектуальних) методах видобутку інформації принциповою відмінністю є повне відсторонення людини від процесу вилучення даних – сторінки аналізуються і інформація витягується автоматично без будь-якої участі оператора [21].

Очевидно, у зв'язку зі зростанням інтересу до технологій big data і data mining, найбільш затребуваним є саме останній підхід – повністю автоматичне вилучення даних. Однак буває, що при використанні автоматичних (а іноді і напівавтоматичних) методів видобутку веб-контенту, часто можна зіткнутися з деякими проблемами, мова про які йтиме нижче.

3.1.1 Проблема розмітки

Проблема вилучення тексту і слабоструктурованих даних з веб-сторінок посилюється повсюдним порушенням і змішанням стандартів і рекомендацій по верстці веб-сторінок. Це може бути викликано як цілями конкретного сайту

(для газети – формування потрібної думки за допомогою спеціального оформлення та компонування матеріалу), так і простою недбалістю та помилками в коді сторінок сайту (це може бути пов'язано з недостатньою кваліфікацією осіб, відповідальних за його створення або функціонування). Ситуація ще більше посилюється досконалістю веб-браузерів, які коректно відображають дані сторінки, проте спеціалізовані парсери зазвичай не здатні справлятися з даною проблемою.

Наприклад , є фрагмент коду сторінки одного з сайтів, що ілюструє останній абзац:

```
…який їде
 <http://www.***.ua/****/*****/> в потрібне</a> вам місце…
```

За тегом, що закриває , можна припустити, що пропущено фрагмент «A HRef=» після першої відкриваючої кутової дужки.

Також можливі помилкові фрагменти коду:

```
…що еквівалентно z=x-1, z(-<span style="face="Times New
Roman">&#8734; </span>,-1]. …
```

Тут помітна спроба змішування форматування як за допомогою стилів, так і за допомогою тега <FONT>. Даний приклад складений трьома лапками, які повинні вживатися суворо попарно. Більшість парсерів або вважають третю лапку відкриттям і пропускають весь текст до кінця сторінки (намагаючись помістити його в значення атрибуту тега), або зупиняються на кутовій дужці, що закриває тег.

У багатьох текстах, що заміщають зображення, можна зустріти подібні багаторядкові конструкції:

```
<img src="…" alt="перший рядок<br/>другий рядок<br¶/>інші¶ рядки">
```

Тут є наступні проблемні моменти:

- використання розмітки HTML всередині значення атрибута;
- кілька «жорстких» переходів рядка (позначені символом «¶»);
- змішання стандартів коду – HTML і xHTML.

Перша проблема є принциповою і поведінка парсерів HTML при зустрічі з нею передбачити неможливо. Найчастіша поведінка в таких випадках – пропадає частина сторінки між «відкриваючим» проблемним тегом і таким же проблемним тегом у наступної картинки, який розглядається як «закриваючий». Подібна поведінка характерна для парсерів HTML на основі кінцевих автоматів (тобто практично всіх) – зустрівши спірну конструкцію, він переходить у стан очікування «закриваючої» конструкції, а коли знаходить – продовжує розбір, при цьому з тексту вичленовується неправильний фрагмент.

Проілюструємо проблему розмітки на прикладі проблемних моментів в тегах зображень. Припустимо, користувач хоче завантажити на комп'ютер обкладинки книг з книжного інтернет магазину, проте в тегах зображень, що їх містять, наявні помилки, що унеможливлює задачу. Розглянемо, що користувач може зробити в таких випадках.

Щоб отримати повний досвід роботи з BeautifulSoup, перш за все потрібно встановити парсер [22]. В Інтернеті можна часто зустріти рекомендації з використання lxml-парсера для підвищення швидкості парсингу, але можна також скористатися іншими рішеннями (наприклад, вбудованим у Python html.parser).

Наступною річчю, потрібною для початку сканування в Інтернеті, є бібліотека запитів requests, за допомогою якої ми можемо запитувати веб-сторінки з веб-сайтів. Імпортуємо потрібні бібліотеки в середі Jupyter Notebook, такі як requests і BeautifulSoup:

```
import requests
from bs4 import BeautifulSoup
```

Для прикладу попрацюємо з першою сторінкою списку бестселерів з книжного інтернет-магазину і спробуємо завантажити обкладинки книг на комп'ютер. Для цього надсилаємо запит на першу веб-сторінку сайту. Порядок дій наступний: ми зберігаємо URL-адресу, яку хочемо отримати, у змінну http, потім запитуємо URL-адресу (requests.get(http)) і зберігаємо HTML-вміст сторінки у спеціальній змінній html.

```
http = "https://book-ye.com.ua/catalog/vydavnytstva/filter
/top-is-true/"
response = requests.get(http)
html = response.content
```

Для того, щоб проілюструвати проблему розмітки, внесемо зміни в HTML контент сторінки за допомогою методів роботи зі строковими даними. Наприклад, перейменуємо класс «product__media» на «product_media» тегу <img>, а також добавимо тег
 переносу рядка в атрибут «scr», що містить посилання на зображення з обкладинками книг. Після цього передамо навмисне зіпсовані дані в об’єкт BeautifulSoup за допомогою наступних рядків коду:

```
html = html.replace(b'class="product__media"',
b'class="product_media"')
html = html.replace(b'src="', b'src="<br>')

soup = BeautifulSoup(html, "html.parser")
```

Тож, ми імпортували BeautifulSoup у вигляді об'єкту, після чого першим параметром методу BeautifulSoup() передали змінну html, що містить змінений HTML-контент із отриманої URL-адреси бестселерів; другий параметр (“html.parser”) – це парсер, який використовується на змінній html. Через пошкодження даних зображень, обкладинки книг не відображаються, як показано на рисунку 3.1.

Рисунок 3.1 – Пошкоджені дані тегу <img> з зображеннями обкладинок

Далі переходимо до завантаження зображень обкладинок на комп'ютер. Для цього запустимо наступний код лістингу 3.1:

Лістинг 3.1 – Видобуток зображень обкладинок бестселерів на комп’ютер

```
file_names = []
url = 'https://book-ye.com.ua'

image_tags = soup.find_all('img', class_="product__media")

if (len(image_tags)!=0):
    for image_tag in image_tags:
        link = url+image_tag['data-src']
        with open(basename(link), "wb") as f:
            f.write(requests.get(link).content)
        file_names.append(basename(link))
else:

    image_tags = soup.find_all('img', class_="product_media")

    for image_tag in image_tags:

        link = url+image_tag['data-src']
        try:
            with open(basename(link), "wb") as f:
                f.write(requests.get(link).content)
            file_names.append(basename(link))

        except:

            link_fixed = link.replace('<br>', '')

            with open(basename(link_fixed), "wb") as f:
                f.write(requests.get(link_fixed).content)
            file_names.append(basename(link_fixed))
```

Перш за все намагаємося знайти всі теги <img>, що містять в атрибутах клас «product__media». Для перевірки того, що знайшовся хоча б один такий тег, ставимо умову, що список image_tags, що містить в собі результати пошуку, має хоча б один елемент. Якщо це так, то завантажуємо зображення на комп’ютер за допомогою методу write(). Якщо умова не виконується, значить назва класу вказана неправильно, тому програма перейде до обробки блоку виключення, де списку image_tags буде вказано знайти посилання на

зображення з елементу <img> з іншою назвою класу. В наступному блоці try, розміщеному в циклі з кількістю ітерацій, що дорівнює кількості посилань, намагаємося зберегти кожне зображення за посиланням на комп'ютер. Проте, через наявність в атрибуті «str» конструкції
, збереження зображень не є можливим, адже посилання в такому випадку псуються і будуть переводити на неіснуючі адреси. В блоці обробки виключень except проводиться вирізання конструкції
 з посилань за допомогою методу replace(), після чого збереження зображень проходить успішно.

Викликавши методи для зчитування та відображення завантажених зображень обкладинок, отримаємо результат на рисунку 3.2.

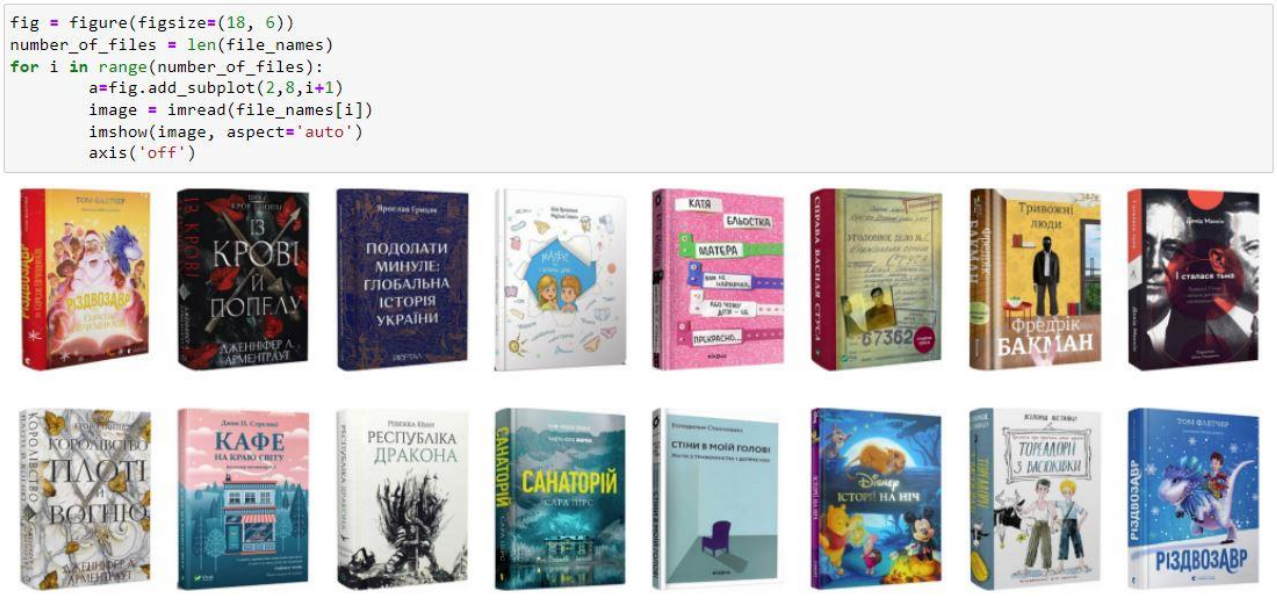


Рисунок 3.2 – Відображення завантажених зображень з обкладинками бестселерів

3.1.2 Проблема навігації

Під проблемою навігації (navigation problem) мається на увазі проблема пошуку та аналізу на веб-сайті сторінки з отримуваною інформацією. При цьому необхідно визначити, чи вся цікава інформація знаходиться на сторінці і

виділити посилання на інші її частини (зазвичай робиться навпаки – якщо немає посилань на інші сторінки – значить, інших сторінок немає). Основна проблема навігації полягає у використанні розробниками коду веб-сторінок сучасних або незвичайних технологій, а також просто порушення наступних негласних правил проектування сайтів.

Одне із таких правил пов’язано з навігацією на сайті за допомогою коду JavaScript. Очевидно, що при цьому для переходу на іншу сторінку сайту потрібно виконання коду, за допомогою якого здійснюється навігація, в той час як парсер, який витягує потрібні дані, зазвичай за структурою є набагато простішим за браузер і не виконує скриптів (як правило, до парсеру також пред'являються певні требування по швидкодії, що автоматично виключає будь-яку необхідність виконання скриптів парсером).

Інша сторона проблеми пов’язана з завантаженням контенту за допомогою технології AJAX з «безкінечною» прокруткою веб-сторінки. Ця проблема доповнює собою попередню – тут не тільки неможливо отримати посилання на наступну сторінку з самої сторінки, але ситуація усугубляється періодичним завантаженням контенту за запитом з цієї сторінки. Проблема є, в більшості випадків, фундаментальною і не вирішується без виконання коду, що здійснює показ необхідного вмісту. Цей підхід також іноді використовується для захисту веб-сайтів від несанкціонованого копіювання, проте його основне призначення заключається в збільшенні зручності використання сайту для користувачів. Прикладом використання цієї технології є прокручування списку відео через смугу прокручування сторінки браузером на сайті-платформі «YouTube».

Можна розглянути дану проблему на прикладі відомого сайту з каталогами товарів [23]. Якщо уважно проаналізувати поведінку веб-сторінки з товарами, можна помітити, що контент на ній підвантажуються динамічно, тобто при прокручуванні сторінки вниз показується новий набір товарів, як показано на рисунку 3.3.

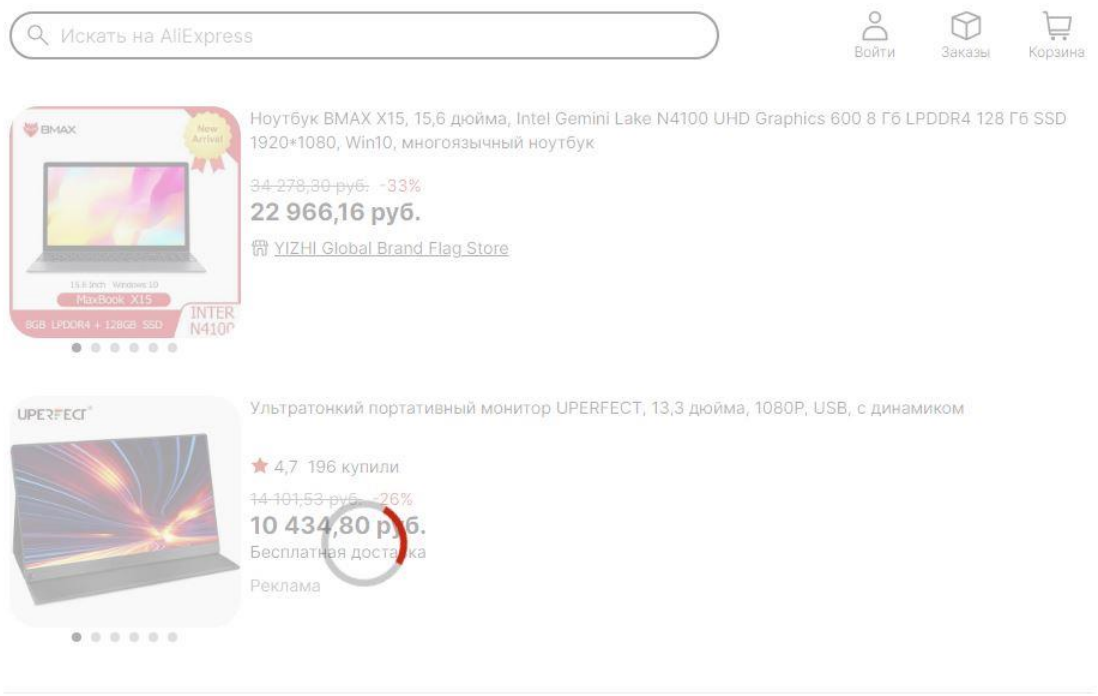


Рисунок 3.3 – «Безкінечна» прокрутка AJAX сторінки з товарами

Для таких випадків існує програмне рішення – бібліотека Selenium програмної мови Python, яка запускає двигун браузера і емулює поведінку людини. Для прикладу зберемо в список назви товарів і їх ціну, але вже з використанням Selenium.

Перш за все потрібно підключити бібліотеку webdriver. Selenium WebDriver − це веб-фреймворк, який дозволяє виконувати кросбраузерні тести. Цей інструмент використовується для автоматизації тестування веб-додатків для перевірки того, що вони працюють належним чином. Для його роботи необхідна наявність драйвера браузера, в якому буде проводитися тестування. У даному випадку був використаний драйвер браузера Opera.

У наступному фрагменті коду ми як і раніше get-запитом з параметрами викликаємо бажану сторінку браузера і завантажуємо дані, після чого у нас з'являється об'єкт класу bs з бібліотеки BeautifulSoup, з яким ми проводимо операції вилучення.

```
from selenium import webdriver
from bs4 import BeautifulSoup as bs
```

```
browser = webdriver.Opera(executable_path='C:\Program
Files\Opera\operadriver\operadriver.exe')
browser.get("https://aliexpress.ru/category/202000104/laptops.html
?g=n&page=3&spm=a2g0o.category_nav.1.220.464a5d8bsqZ2")
```

Після виконання коду відкриється вікно браузера з заданим URL. Однак буває, що при першому вході на сайт потрібно вказати налаштування регіону, мови та валюти (рисунок 3.4) для відображення опису товарів потрібною мовою і цін в потрібній валюті. Це також вирішить проблему постійного переадресування користувача на головну сторінку під час кожного сеансу входу браузера за вказаним посиланням.

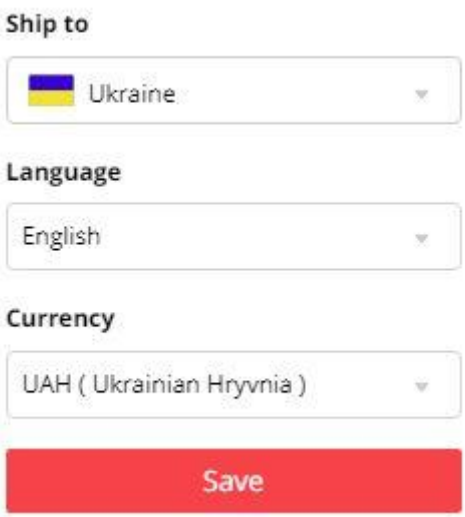


Рисунок 3.4 – Вибір регіону, мови та валюти для каталогу с товарами

В даному випадку скористаємося методом Session бібліотеки requests, в яку можна передати cookies в якості параметра. Після цього код буде виглядати наступним чином:

```
cookie = {'name': 'aep_usuc_f', 'value':
'isfm=y&site=rus&c_tp=UAH&isb=y&region=UA&b_locale=uk_UA',
'domain': '.aliexpress.com'}
browser.add_cookie(cookie)
```

Для того щоб почати завантажувати дані з динамічної веб-сторінки,

необхідно виконати виконати деякі команди для роботи движка браузера, які символізують переміщення курсору вниз. Функція sleep() з бібліотеки time затримує виконання поточного потоку на задану кількість секунд. В даному випадку скрипт, наведений в лістингу 3.2, виконує прокручування сторінки вниз до самого кінця, чекає загрузки нового контенту 3 секунди, після чого повторює описані дії, допоки в категорії товару не закінчиться весь контент.

Лістинг 3.2 – Скрипт для автоматичного прокручування сторінки вниз

```
import time

lenOfPage = browser.execute_script("window.scrollTo(0,
document.body.scrollHeight);var
lenOfPage=document.body.scrollHeight;return lenOfPage;")
match=False
while(match==False):
      lastCount = lenOfPage
      time.sleep(3)
      lenOfPage = browser.execute_script("window.scrollTo(0,
document.body.scrollHeight);var
lenOfPage=document.body.scrollHeight;return lenOfPage;")
      if lastCount==lenOfPage:
            match=True
```

Тепер ми дійшли до кінця сторінки і можемо зібрати дані для роботи бібліотеки BeautifulSoup. Приступимо до збору даних про товари. Для цього імпортуємо бібліотеку pandas та створюємо новий dataframe з двома колонками: «Назва товару» і «Ціна».

Лістинг 3.3 – Видобуток даних з використанням бібліотеки pandas

```
import pandas as pd

source_data = browser.page_source
soup = bs(source_data)

models=soup.find_all('div', {'class':['product-
snippet_ProductSnippet__name__1ettdy']})
price=soup.find_all('div', {'class':['snow-
price_SnowPrice__mainM__ugww0l']})
```

```
df = pd.DataFrame({'Назва товару': models, 'Ціна': price})
browser.close()
```

У наведеному вище фрагменті коду ми шукаємо всі елементи , які мають клас product-snippet_ProductSnippet__name__1ettdy, що містить в собі назву товару (рисунок 3.5), а також елементи з класом snow-price_SnowPrice__mainM__ugww0l, щоб зібрати значення ціни для кожного знайденого товару.

Метод close() використовується для закриття поточного вікна браузера, на якому встановлено фокус і у якому Selenium виконує автоматичні тести, однак сеанс WebDriver при цьому залишається активним.

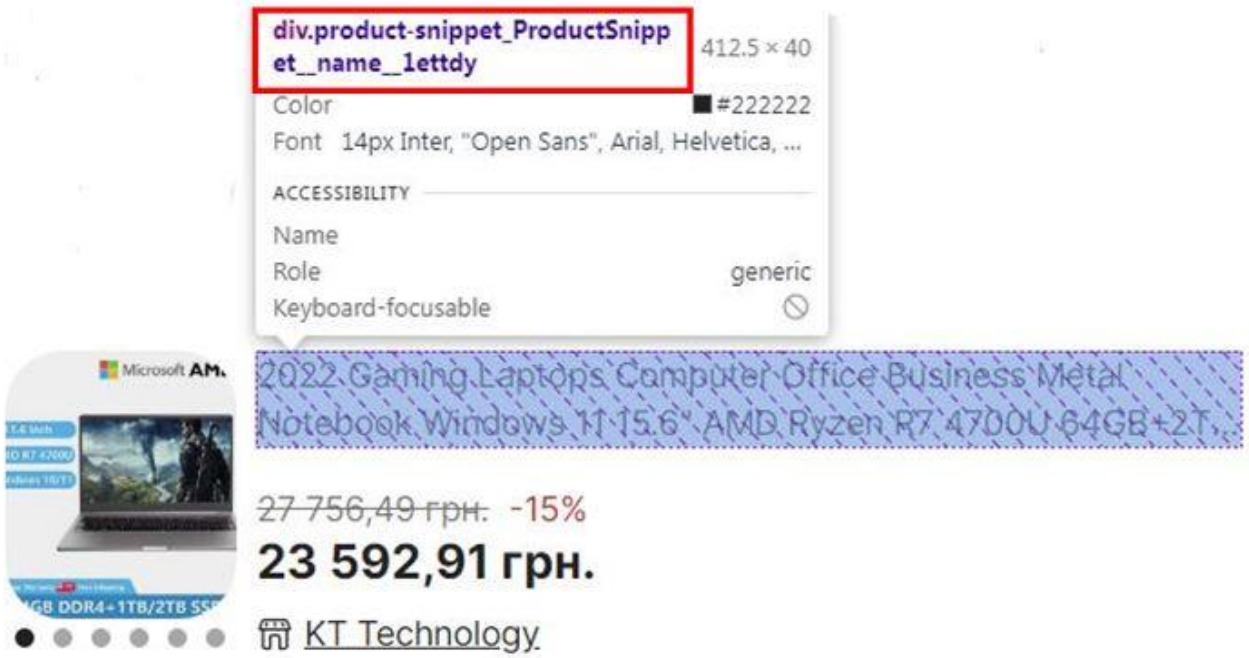


Рисунок 3.5 – Елемент , що містить назву товару

Викликавши команду для перегляду вміст датафрейму з даними про товари, можна отримати наступну інформацію на рисунку 3.6, що сигналізує про успішне вилучення.

	Назва товару	Ціна
0	[Ноутбук Lenovo 82TT0010RU]	[64 385,01 руб.]
1	[Ноутбук Lenovo IdeaPad L3 15ITL6 82HL0039RK 1...	[55 286,00 руб.]
2	[Ноутбук Leap T304 SF20GM6 SF20GM6 11.6"]	[15 499,00 руб.]
3	[Ноутбук Acer Extensa 15 EX215-54-355T NX.EGJE...	[44 292,00 руб.]
4	[Ноутбук Dere R9 Pro, 15,6 дюйма, 16 ГБ ОЗУ, Т...	[17 374,38 руб.]
5	[Переходник NS-A759 для ноутбука Lenovo IdeaPa...	[535,00 руб.]
6	[Ноутбук HP Pavilion 14 14-ec0034ur 4E1A8EA R5...	[83 833,00 руб.]
7	[Ноутбук Acer TravelMate TMP215-52-30CQ 15.6" ...	[50 483,64 руб.]
8	[2022 металлические Игровые ноутбуки Win10, но...	[51 139,71 руб.]
9	[Портативный высокоскоростной мобильный твердо...	[695,00 руб.]

Рисунок 3.6 – Список товарів і їх ціни з сайту-каталогу

Використання у внутрішньому механізмі сторінкового показу не номерів сторінок в даний час майже не зустрічається – зазвичай при навігації використовується послідовна нумерація сторінок (з кроком 1 або більше (наприклад, на форумах за номерами першого повідомлення на сторінці)). Однак зустрічається навігація, наприклад, деяким id або дат.

Аналіз відбуватиметься тим швидше і ефективніше, чим менше буде знайдено дублікатів сторінок з інформацією, що видобувається. Дублікати порушують статистичний розподіл елементів і можуть порушити розпізнавання. Проблема полягає в тому, що багато CMS (системи управління вмістом сайту) надають доступ до основних сторінок за різними адресами.

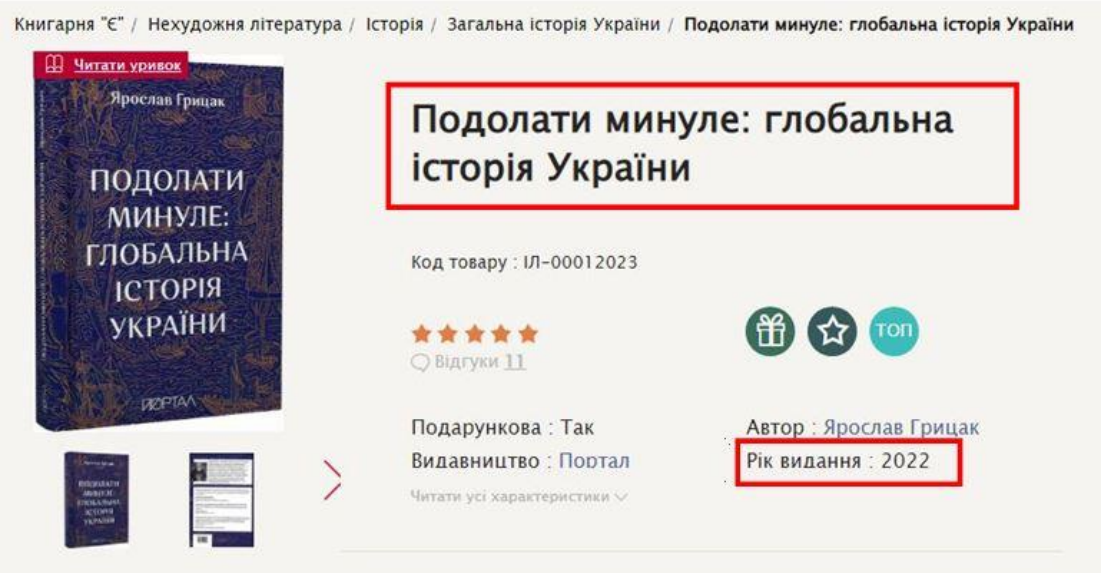
Наступна проблема навігації полягає в розміщенні контенту на декількох сторінках сайту. Наприклад, користувачеві потрібно вилучити дані про назву, ціну, попередню ціну та рік видання книг з сайту-каталогу з книгами, тому він користується парсером для видобутку даних зі сторінки. Але труднощі полягають в тому, що насправді є 7 сторінок бестселерів, кожну з яких потрібно

перебрати, як показано на рисунку 3.7.

Рисунок 3.7 – Розміщення контенту веб-сайту на декількох сторінках

Крім того, деякі з даних, які підлягають видобутку (такі як рік видання), можна отримати лише при переході на окрему сторінку з інформацією про книгу, як показано на рисунку 3.8 а) і б).

а)



б)

Рисунок 3.8 – Розміщення даних про бестселери: а) на сторінці зі списком книг; б) на окремій сторінці з деталями товару

В цілому маємо ситуацію, де потрібно проаналізувати дані про назву, ціну, попередню ціну до знижки та рік видання не 16 книг, розміщених на першій сторінці, а більше 96 книг (6 перших сторінок плюс остання, яка може бути заповнена не до кінця). Крім того, дані про рік видання доступні лише за окремим посиланням, тобто для їх отримання потрібно буде переходити на сторінку кожної окремої книги.

Для початку роботи з видобутку даних потрібно підключити вже знайомі бібліотеки, після чого отримати URL-адресу для нової сторінки, завантажити сторінку за допомогою модуля requests та перетворити її на HTML-документ за допомогою BeautifulSoup [24]. Код наведено в лістингу 3.3.

Лістинг 3.4 – Попередня підготовка перед видобутком даних

```
import requests
from bs4 import BeautifulSoup
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt

http = "https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-
true/"
response = requests.get(http)
html = response.content
soup = BeautifulSoup(html, "lxml")

page = 1
last_page = int(soup.find_all("a", class_="pagination__item")[-
2].get_text())
```

Змінна page знадобиться надалі для переходу на нові сторінки, тоді як змінна last_page містить в собі номер останньої сторінки, отриманий з класу «pagination__item» елемента <a>, який є елементом навігації в верхній частині сайту.

Проте все ще залишається питання про те, як отримувати посилання на наступні сторінки. Рішення полягає в перевірці того, що відбувається в URL-адресі, коли ми їх перемикаємо. Наступна URL-адреса є адресою першої сторінки: http_page = https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/. Перейшовши на другу сторінку, можна помітити, що URL-адреса зміниться на наступну: https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=2.

Єдина відмінність полягає в тому, що ?PAGEN_1=2 було додано до основної URL-адреси. Якщо ми відвідаємо третю сторінку, посилання зміниться знову: https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=3. Таким чином, змінивши число після ?PAGEN_1=, можна перейти на будь-яку сторінку за бажанням.

Проте при першому заході на початкову сторінку, в ній не було частини ?PAGEN_1=<число>. Це пояснюється тим, що два посилання: https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=1 і https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/ – це та сама сторінка з однаковими результатами пошуку книг. Це може стати в нагоді, як надійне рішення, яке ми можемо використовувати для навігації між веб-сторінками,

змінюючи URL-адресу.

Частина URL-адреси зі знаком «?» означає початок так званого рядка запиту. Усе, що йде після «?» – це сам рядок запиту, який містить пари ключ-значення. У даному випадку сторінка є ключем, а номер, який ми їй призначаємо, є її значенням. Присвоївши певний номер сторінці, ми можемо запросити сторінку бестселерів, яка відповідає цьому номеру.

Скориставшись цим, напишемо код в лістингу 3.4 для отримання назв, цін та років публікації бестселерів. Щоб сканувати кілька сторінок, використаємо цикл while і параметри page в URL-адресах.

Лістинг 3.5 – Видобуток даних про бестселери

```
while page != last_page+1:
      http_page = f"https://book-
ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1={page}"

      response = requests.get(http_page)
      html = response.content
      soup = BeautifulSoup(html, "lxml")

      books_on_Page = len(soup.find_all("a",
class_="product__name"))

      for i in range(0, books_on_Page):
            url = 'https://book-ye.com.ua' + soup.find_all("a",
class_="product__name")[i]['href']

            response = requests.get(url)
            soup1 = BeautifulSoup(response.text, "html.parser")

            book = {}

            title = soup.find_all("a", class_="product__name")[i]
            price = soup.find_all("div", class_="product__price-
current")[i]
            price_old = soup.find_all("div",
class_="product__price-old")[i]
            publ_year = soup1.find("meta",
itemprop="copyrightYear", content=True)['content']

            years_list.append(publ_year)
            prices_list.append(price)
            prices_old_list.append(price_old)
```

```
            book["Назва"] = title.get_text(strip=True)
            book["Ціна"] = float(price.get_text().replace(' ',
'').replace('грн', ''))
            try:
                book["Попередня ціна"] =
float(price_old.get_text().replace('грн', ''))
            except:
                book["Попередня ціна"] = book["Ціна"]
            book["Рік видання"] = publ_year
            best_book.append(book)

      years_series = pd.Series(years_list)
      page = page + 1
```

Використання циклу while та передача в ньому номеру сторінки в кінець змінної http_page при кожній ітерації, генерує посилання на сторінки каталогу з даними бестселерів (рисунок 3.9) [24].

```
page = 1
while page != last_page+1:
      http_page = f"https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1={page}"
      print(http_page)
      page = page + 1
```

```
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=1
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=2
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=3
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=4
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=5
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=6
https://book-ye.com.ua/catalog/vydavnytstva/filter/top-is-true/?PAGEN_1=7
```

Рисунок 3.9 – Генерація посилань на сторінки з даними бестселерів

Цикл for потрібен для генерації посилань на окрему сторінку для кожної книги. Для цього потрібно попередньо сканувати кількість книг на сторінці і передати в змінну books_on_Page, таким чином отримавши більше 96 посилань. Списки title, price, price_old і publ_year вміщають в себе назву, ціну, попередню ціну та дату публікації відповідно для кожної книги. Використання словника book, який приймає дані зі списків, зручне тим, що містить в собі одночасно інформацію про всі книги з каталогу сайту, при чому ці дані пов’язані і однозначно відповідають один одному.

Викликавши команду для перегляду вмісту датафрейму з даними про бестселери, можна отримати наступну інформацію на рисунку 3.10, що свідчить про успішне вилучення.

	Назва	Ціна	Попередня ціна	Рік видання
0	Різдвозавр та список Нечемнюхів	179 грн	220 грн	2021
1	Подолати минуле: глобальна історія ...	680 грн	708 грн	2022
2	Малечі про інтимні речі	219 грн	277 грн	2021
3	Кров і попіл. Книга 1. Із крові й п...	276 грн	378 грн	2022
4	Матера вам не наймичка, або Чому ді...	215 грн	246 грн	2021
...	...	...	...	...
93	Музей покинутих секретів	259 грн	403 грн	2020
94	Ожинова зима	149 грн	200 грн	2022
95	Володар перснів. Повернення короля	255 грн	402 грн	2020
96	Володар перснів. Дві вежі	231 грн	374 грн	2020
97	Фіалки в березні	149 грн	166 грн	2022

98 rows × 4 columns

Рисунок 3.10 – Список бестселерів з інформацією про них

Існує окремий випадок вищеописаної проблеми розміщення контенту на декількох сторінках, в якому розпізнавання посторінкових посилань може некоректно працювати з їх незначною кількістю. Наприклад, стандартна навігація для двох сторінок може мати шість сторінкових посилань приблизно наступного виду:

<< < 1 2 > >>,

де:

«2» – поточна сторінка, а також гіперпосилання на поточну сторінку,

«1» – гіперпосилання на першу сторінку, причому перша мається на увазі в сенсі «попередня перед поточною»,

«<<» – гіперпосилання для переходу до першої сторінки,

«<» – гіперпосилання для переходу до попередньої сторінки,

«>>» – гіперпосилання для переходу до останньої сторінки,

«>» – гіперпосилання для переходу до наступної сторінки.

Таким чином, шість посилань на дві сторінки (1, 1, 1, 2, 2, 2) здатні створити складність для розпізнавання елементів посторінкової навігації. Зазвичай в таких випадках доводиться користуватися методом масового відвідування даних всіх можливих сторінок і подальшого їх аналізу з постобробкою.

На практиці можливе створення парсера на основі ActiveX-компонента Internet Explorer (або компонентів модуля QtWebKit), де парсер при цьому буде отримувати можливості відповідного браузера, у тому числі і повний доступ до динамічно створюваного та/або завантажуваного контенту. Можливе також написання розширення до браузеру з аналогічною метою.

Деякі сучасні CMS здатні надавати доступ до потрібної інформації по прямим посиланням, через зміст сайту та іншими способами, тобто не тільки через сторінку, що використовує даний проблемний підхід.

Крім цього на практиці зустрічаються й інші проблеми – сайт повністю або цілком може бути зроблений з використанням технології Flash, мати обмеження на кількість запитів в інтервал часу (у секунду, в хвилину, в годину), обмеження на кількість запитів з однієї IP-адреси, обмеження доступу до всіх або окремих сторінок сайту. Загальним, але дієвим рішенням даної проблеми слід вважати примусове обмеження кількості з'єднань з сайтом (не більше 3-х в секунду).

3.1.3 Проблема розпізнавання вилучених даних

Проблема розпізнавання даних і структур (data extraction problem) полягає в необхідності визначення ділянок на веб-сторінці, що містять інформацію і структуру даних, що видобуваються.

Найбільш часто, але не завжди, дане завдання ставить за мету пошук

повторюваних структур даних. У найпростішому випадку веб-сторінку слід подати у вигляді дерева DOM-елементів і шукати вузли, що мають однакову структуру до-чорних вузлів. Існують також і інші способи, наведені нижче.

На основі аналізу графічного представлення веб-сторінки: це надзвичайно ресурсомісткий, але ефективний підхід. На практиці не складно знайти сайти з проблемною розміткою, яка некоректно відображається і, відповідно, погано піддається аналізу. Також на окремих групах сайтів нормою стали спливаючі модальні вікна (як з рекламою, так і з пропозицією використовувати, наприклад, додатки та соціальні послуги даного сайту), що закривають собою контент.

Пошук на основі семантичної розмітки і мікроформатів: даний спосіб відповідає початковим цілям створення Всесвітньої павутини, але на практиці його реалізувати важко, адже семантична розмітка в даний час масово використовується практично з протилежними цілями – для підвищення рейтингу в пошукових системах, а не для смислової розмітки контенту.

Текстовий аналіз: даний підхід передбачає виділення елементів по їх вмісту з урахуванням HTML-розмітки, що обрамляє. Наприклад, досить часто в тексті зустрічаються власні імена. Відповідно, база даних з іменами, прізвищами, назвами компаній, країн і населених пунктів здатна значно підвищити якість виділення інформативних текстових блоків. Є і інший варіант: відомо, що в HTML є потокові елементи, а є блокові (керуючі взаємним положенням елементів при їх відображенні). Можна організувати структуру сайту так, щоб вся верстка формувалася за допомогою блокових елементів, а все форматування – потоковими елементами. Найбільш рідкісні винятки є специфічними і їх можна ігнорувати при вилученні даних.

3.1.4 Проблема забезпечення однорідності даних

Проблема забезпечення однорідності даних полягає у забезпеченні однорідності інформації, яка може бути представлена на веб-сторінці з деякою

варіативністю атрибутів. Для прикладу можна розглянути наступний складний випадок: нехай існує каталог статей, де кожна стаття представлена файлом з нею, назвою, автором, анотацією, ключовими словами тощо. Якщо значна кількість полів каталогу не буде заповнена, інформацію з них не буде виведено. Зрозуміло, що при коректному вилученні даних відсутні поля слід залишити порожніми, але головна проблема не в цьому.

Основна складність тут полягає в можливому збої визначення кількості вилучених полів для кожного запису, тому, можливо, що при кластеризації дані будуть згруповані неправильним способом (наприклад, статті з анотаціями будуть утворювати один набір даних, а статті з анотаціями і авторами - інший, що, у даному випадку, неправильно), як показано на рисунку 3.11.

File	Author <string>	Date <date_type>	Size (MB) <float>
file1.doc	J. A. Saimon	15.10.2020	3,16
L. M. Derek	17.10.2020	2,20	<NONE>
file3.doc	A. Richard	20.11.2021	2,83
file4.doc	F. L. March	1,6	<NONE>

Рисунок 3.11 – Неправильна кластеризація даних

Крім того, деякі дані часто слід нормалізувати окремими алгоритмами: наприклад, дані номеру телефона можуть бути записані як «тел. +380 XX XXX-XX-XX», «0XXXXXXXXX», «+38 (0XX) XXX-XX-XX», та іншими способами.

Аналогічно – дати і час, населені пункти (наприклад, "м. Санкт-Петербург", "Санкт-Петербург", "Пітер" та ін).

3.1.5 Проблема об’єднання даних

Проблема об'єднання даних полягає в наступному: конкретна одиниця інформації (запис) може бути представлена на сайті (або навіть сторінці) неодноразово, тому після закінчення виділення необхідно забезпечити видалення дублікатів. Це завдання легко можна вирішити після закінчення вилучення даних, проте парсинг є більш ресурсомісткою операцією у порівнянні з прийняттям рішення по закінченні, тому оптимальнішим рішенням буде виявляти дублікати до його початку. Проте, не завжди в принципі можна визначити дублікати сторінок. Наприклад, в деяких інтернет-магазинах радіодеталей існує безліч повністю однакових записів, і тільки при відкритті конкретного товару можна визначити, що, наприклад, йдеться про різні партії від різних постачальників.

Після здійснення даних етапів інформація з веб-сторінки переходить в систематизовану форму, придатну для імпорту в базу знань.

Однак не завжди слід зосереджуватися повністю на автоматичному вилученні інформації з веб-сторінок. У зв'язку з повсюдним поширенням одних і тих самих CMS зі стандартними модулями і розширеннями, всього 37 груп складних правил здатні коректно обробити більше 99% україномовних сайтів в напівавтоматичному режимі.

3.2 Візуалізація даних

Однією з переваг мови Python як інструменту видобутку веб-контенту є наявність в ній зручних засобів візуалізації, за допомогою яких можна показати результуючі дані в різноманітних формах. Наприклад, помістивши дані з об’єкту-словника на рисунку 3.12 а) в датафрейм-об’єкт з бібліотеки pandas на

рисунку 3.12 б), інформацію про книги можна показати в більш зрозумілому вигляді, порівняно з представленням у вигляді словника з даними.

```
[{'Назва': 'Різдвозавр та список Нечемнюхів',
  'Ціна': '179 грн',
  'Попередня ціна': '220 грн',
  'Рік видання': '2021'},
 {'Назва': 'Подолати минуле: глобальна історія ...',
  'Ціна': '680 грн',
  'Попередня ціна': '708 грн',
  'Рік видання': '2022'},
 {'Назва': 'Малечі про інтимні речі',
  'Ціна': '219 грн',
  'Попередня ціна': '277 грн',
  'Рік видання': '2021'},
 {'Назва': 'Кров і попіл. Книга 1. Із крові й п...',
  'Ціна': '276 грн',
  'Попередня ціна': '378 грн',
  'Рік видання': '2022'},
```

а)

	Назва	Ціна	Попередня ціна	Рік видання
0	Різдвозавр та список Нечемнюхів	179 грн	220 грн	2021
1	Подолати минуле: глобальна історія ...	680 грн	708 грн	2022
2	Малечі про інтимні речі	219 грн	277 грн	2021
3	Кров і попіл. Книга 1. Із крові й п...	276 грн	378 грн	2022
4	Матера вам не наймичка, або Чому ді...	215 грн	246 грн	2021
...	...	...	...	...
93	Музей покинутих секретів	259 грн	403 грн	2020
94	Ожинова зима	149 грн	200 грн	2022
95	Володар перснів. Повернення короля	255 грн	402 грн	2020
96	Володар перснів. Дві вежі	231 грн	374 грн	2020
97	Фіалки в березні	149 грн	166 грн	2022

98 rows × 4 columns

б)

Рисунок 3.12 – Візуалізація даних про бестселери: а) у вигляді словника; б) з використанням бібліотеки pandas

Наступною бібліотекою візуалізації, що асоціюється з виведенням графіків, діаграм та гістограм, є бібіотека matplotlib.

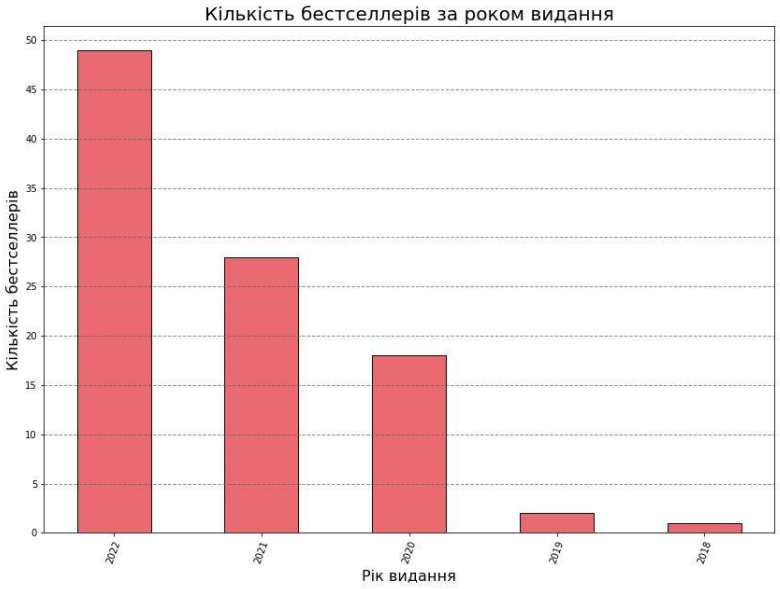


Рисунок 3.13 – Візуалізація кількості бестселерів, випущених за певний рік

На рисунку 3.13 показано використання бібліотеки matplotlib для відображення статистики про кількість бестселерів за певним роком видання. Як можна помітити, найбільша кількість бестселерів припадає на 2022 (поточний) рік, і це не дивно, адже усі бестселери, випущені за поточний рік, розміщуються на перших декількох сторінках каталогу, а книги більш старого року випуску розміщуються або на останніх сторінках, або випадають зі списку взагалі. Таким чином, використання бібліотек matplotlib та pandas є зручним способом візуалізації даної статистики.

Також в мові Python можна легко вивести інформацію зразу по декільком вимірам даних. Наприклад, інформацію про знижки та рік видання для кожної окремої книги, яка представлена у вигляді тривимірного графу, як на рисунку 3.14. Таке представлення буде нагадувати OLAP-куб, але без використання сховищ даних та сторонніх програмних засобів.

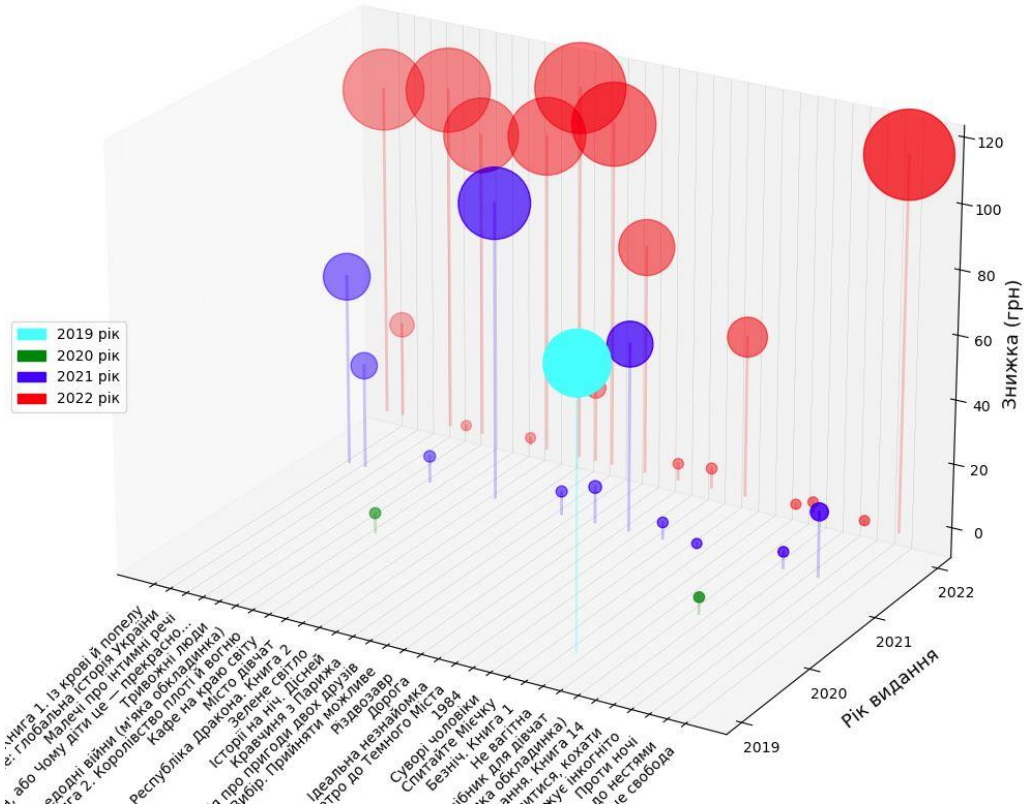


Рисунок 3.14 – Багатовимірна візуалізація даних по бестселерам

Як інструмент аналізу та візуалізації інформації, Python краще використовувати для невеликих проектів, або для проектів з конкретним специфічним завданням (наприклад, пошук найдешевшого товару з каталогу певного сайту). Для масштабних проектів з великою кількістю даних, інформацію краще зберігати в сховищах, а для її аналізу використовувати спеціально призначені для цього програмні продукти.

3.3 Висновки

В даному розділі були перелічені проблеми автоматичного вилучення даних із веб-сайтів, а також способи їх вирішення. Це такі проблеми, як: проблема розмітки; навігації; розпізнавання, об’єднання та забезпечення однорідності даних. Проблема розмітки може бути вирішена внесенням поправок в код елементів HTML за допомогою методів роботи зі строковими даними. Проблема навігації, коли інформація розміщується на декількох сторінках, вирішується використанням циклів для отримання списку посилань на кожну окрему сторінку. Якщо це сторінка з динамічною підгрузкою контенту, використовується бібліотека Selenium мови програмування Python, що може імітувати прокручування сторінки людиною.

Також мова програмування Python може виступати не тільки як інструмент вирішення проблем автоматичного вилучення контенту за допомогою спеціально призначених для цього програмних продуктів, але і як інструмент візуалізації кінцевих даних.

ВИСНОВКИ

Інтернет – це величезне сховище інформації та знань, доступних у мережі. Проблема видобутку цінної та корисної інформації з Інтернету полягає в складності, неструктурованості та обсязі веб-сторінок, і ситуація ускладнюється тим, що з часом їх кількість невпинно зростає. Таким чином, для вилучення необхідних даних та знань із вмісту веб-сторінки доводиться застосовувати різноманітні методи, способи і інструменти видобутку веб-контенту. У даній роботі були розглянуті ряд таких інструментів та методів для видобутку веб-контенту з мережі, перелічені їх особливості, переваги та недоліки, проведено порівняльний аналіз.

У другому розділі кваліфікаційної роботи були розглянуті програмні засоби видобутку веб-контенту, включаючи їх порівняння за двома групами: комерційні та некомерційні. Таке розділення було обрано з метою показати, що в залежності від цілей, які ставить користувач під час видобутку даних, некомерційні програмні засоби не поступаються в функціоналі комерційним засобам веб-майнінгу. Результати порівняння програм за декількома критеріями були занесені в таблицю. Також у другому розділі була розглянута мова програмування Python як інструмент автоматичного видобутку веб-контенту в невеликих за масштабом проектах. Результати порівняння бібліотек Python, таких як BeautifulSoup, Scrapy і Selenium також були систематизовані та занесені до таблиці.

В третьому розділі роботи наведено перелік проблем автоматичного вилучення тексту та слабоструктурованих даних із веб-сайтів, значна кількість яких пояснюється простим порушенням стандартів та рекомендацій щодо верстки веб-сторінок і розмітки HTML, неоднорідністю інформації або наявністю дублікатів даних. Проблема навігації, коли інформація розміщується на декількох сторінках або на сторінці з динамічною підгрузкою контенту, вирішується використанням бібліотек мов програмування, здатних імітувати

поведінку людини (прокручування сторінки, натискання клавіш і т.і.).

Ефективність, масштабованість, продуктивність, оптимізація та можливість виконання в режимі реального часу є ключовими критеріями, які стимулюють розробку багатьох нових алгоритмів інтелектуального аналізу даних [2]. Аналіз і теоретичний огляд запропонованих інструментів показали, що ефективність алгоритмів веб-майнінгу залишають за собою місце для вдосконалення. Наприклад, процес розпаралелювання величезного обсягу роботи алгоритмів майнінгу веб-даних міг би покращити продуктивність у майбутньому [25]. Алгоритми розпаралелювання спочатку розбивають дані на частини, після чого кожна частина обробляється паралельно шляхом пошуку шаблонів, допоки не об’єднаються в кінці. Процес розпаралелювання є однією з можливих рекомендацій на майбутнє, оскільки об’єм веб-даних в Інтернеті зростає безперервно з високою швидкістю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Філенко В. П. Методи та інструменти видобутку веб-контенту. *Проблеми інформатизації* : тези доп. X міжн. наук.-техн. конф., 24 – 25 лист. 2022 р. / ЧДТУ, ВА ЗС АР, УТіГН, НТУ “ХПІ”, ХНУРЕ, "ПД ПКНДІ АП", 2022. С. 93.

2. Han, J., Kamber, M., Pei, J. Data Mining: Concepts and Techniques. Third edition. Morgan Kaufmann Publishers, 2006.

3. Mebrahtu, A., Srinivasulu B. Web Content Mining Techniques and Tools. International Journal of Computer Science and Mobile Computing. Vol. 6, Issue 4, April 2017, pp. 49-55.

4. Sinha, A., Raj, N., Haque, S., Haque A., Singh, N. Web Content Mining: Tool, Technique & Concept. IOSR Journal of Computer Engineering, Volume 18, Issue 6, Ver. VI (Nov.-Dec. 2016), pp. 57-60.

5. Kosala, R., Blockeel, H. Web mining Research: A survey. SIGKDD, Volume 2, Issue 1, 2000, pp. 1-15.

6. Sharda D., Chawla, S. Web Content Mining Techniques: A Study. International Journal of Innovative Research in Technology & Science.

7. Pol, K., Patil, N., Patankar, S., Das, C. A Survey on Web Content Mining and extraction of Structured and Semistructured data, in First International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, July 16-18, 2008.

8. Johnson, F., Gupta, S. Web Content Mining Techniques: A Survey. International Journal of Computer Applications (0975–888), Volume 47 – No.11, June 2012.

9. Chidlovskii, B., Ragetli, J., Rijke, M. Automatic Wrapper Generation For Web Search Engines, in Web-Age Information Management, First International Conference (WAIM 2000), Shanghai, China, June 21-23, 2000.

10. Oskouei, R., Hojati, Z. A Comprehensive Comparison between Web

Content Mining Tools: Usages, Capabilities and Limitations, in International Congress of Electrical Engineering Computer Science & Information Technology, Iran, August 2015.

11. Web crawling vs web scraping [Електронний ресурс] – Режим доступу до ресурсу: https://www.zyte.com/learn/difference-between-web-scraping-and-web-crawling/. – Дата доступу: 22.09.2022

12. Easy Web Extract Review [Електронний ресурс] – Режим доступу до ресурсу: http://scraping.pro/easy-web-extract-review/. – Дата доступу: 22.09.2022.

13. Web Content Extractor Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://www.newprosoft.com/webcontentextractor/documentation/. – Дата доступу: 23.09.2022.

14. Shetty, N., Mowla, S. A Study on Web Mining Tools and Techniques. Journal of Engineering and Applied Sciences 12(2):6135-6142, December 2018.

15. Screen-scraper: Data extraction software and services [Електронний ресурс] – Режим доступу до ресурсу: https://www.screen-scraper.com – Дата доступу: 25.09.2022.

16. Automation Anywhere Manual [Електронний ресурс] – Режим доступу до ресурсу: http://www.automationanywhere.com. – Дата доступу: 25.09.2022.

17. Mozenda [Електронний ресурс] – Режим доступу до ресурсу: http://www.mozenda.com/web-mining-software. – Дата доступу: 25.09.2022.

18. Why Python Is Essential for Data Analysis and Data Science [Електронний ресурс] – Режим доступу до ресурсу: https://www.simplilearn.com/why-python-is-essential-for-data-analysis-article. – Дата доступу до ресурсу: 20.10.2022.

19. Difference between BeautifulSoup and Scrapy crawler [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/difference-between-beautifulsoup-and-scrapy-crawler/. – Дата доступу: 20.10.2022.

20. Selenium vs. Beautiful Soup: A Full Comparison [Електронний

ресурс] – Режим доступу до ресурсу: https://www.blazemeter.com/blog/selenium-vs-beautiful-soup-python. – Дата доступу: 21.10.2022.

21. Asghar, S., Iqbal K. Automated Data Mining Techniques: A Critical Literature Review. International Conference on Information Management and Engineering, April 2009. DOI: http://dx.doi.org/10.1109/ICIME.2009.98.

22. How to Scrape Images with Beautifulsoup4 and Python [Електронний ресурс] – Режим доступу до ресурсу: https://python.plainenglish.io/how-to-scrape-images-using-beautifulsoup4-in-python-e7a4ddb904b8. – Дата доступу: 5.11.2022.

23. How to Dynamically Scrape Multiple Amazon Products Pages Using Beautifulsoup? [Електронний ресурс] – Режим доступу до ресурсу: https://www.xbyte.io/how-to-dynamically-scrape-multiple-amazon-products-pages-using-beautifulsoup.php. – Дата доступу: 7.11.2022.

24. How to Scrape Multiple Web Pages [Електронний ресурс] – Режим доступу до ресурсу: https://data36.com/scrape-multiple-web-pages-beautiful-soup-tutorial/. – Дата доступу: 10.11.2022.

25. Panda, B., Trypathy, S., Sethi, N. A Comparative Study on Serial and Parallel Web Content Mining. Int. J. Advanced Networking and Applications, Volume 7, Issue 5, pp. 2882-2886.