Automated Data Extraction and Analysis of Housing Information

Jack Zhou
Feng Chia University, ISTM
Purdue Computer Engineering
Taichung, Taiwan
lucky368368@gmail.com

***Abstract*—This paper presents an automated system for extracting and analyzing housing information from web sources. The system leverages web scraping techniques, a structured database for storing extracted data, and automated geolocation conversion. The solution is implemented using Python and various libraries including SeleniumBase, BeautifulSoup, TinyDB, and geopy.**

Keywords—Web Scraping, Data Extraction, Geolocation, Housing Information, TinyDB, SeleniumBase, BeautifulSoup

I. Introduction

With the rapid growth of online real estate platforms, there is a need for automated tools to extract and analyze housing data efficiently. This paper introduces a comprehensive system designed to scrape housing information from websites, store it in a structured database, and perform geolocation conversion for spatial analysis. The implementation leverages Python libraries such as SeleniumBase for browser automation, BeautifulSoup for HTML parsing, TinyDB for data storage, and geopy for geolocation services.

II. System Design

The system is composed of several modules, each designed to handle specific aspects of the data extraction and analysis process:

- **Crawl.py:** This module is responsible for web scraping. It uses SeleniumBase to automate browser interactions and BeautifulSoup to parse HTML content. The goal of this module is to navigate through web pages, extract relevant housing information, and pass it on for further processing.
- **Database.py:** This module manages data storage. It uses TinyDB with a custom YAML storage adapter to store the extracted data in a human-readable format. This design choice ensures that the database is easy to inspect and modify manually if needed.
- **Housing.py:** This module defines the Housing class, which encapsulates the logic for extracting housing data from HTML elements. It includes methods to parse relevant information such as address, property name, price, and bed count, and also performs geolocation lookups using the geopy library.
- **Main.py:** This is the main entry point of the system. It integrates the various modules to perform end-to-end data extraction, storage, and analysis. This script orchestrates the workflow, ensuring that each component interacts seamlessly with the others.

III. Implementation

The implementation involves the following steps:

- **Web Scraping**: Web scraping is the first step in the data extraction process. The Crawl.py module uses SeleniumBase to automate interactions with web pages, such as logging into Google Maps and navigating through housing listings. BeautifulSoup is then used to parse the HTML content and extract relevant data fields.
- **Data Storage**: Using TinyDB to store the extracted data in a structured format. A custom YAML storage adapter ensures that the database is human-readable. The custom YAMLStorage class ensures that data is stored in a YAML format, which is both human-readable and easily editable.
- **Geolocation**: The Housing.py module defines the Housing class, which includes methods for geolocation conversion. The geopy library is used to convert addresses to latitude and longitude coordinates.
- **Integration and Workflow**: The Main.py script integrates all the modules to perform end-to-end data extraction, storage, and analysis. It orchestrates the workflow, ensuring that each component interacts seamlessly with the others.

IV. Results and Discussion

The system successfully extracts and stores housing data from the specified web source. The use of a human-readable database format facilitates easy inspection and modification of data. Geolocation conversion adds a spatial dimension to the analysis, enabling visualization and mapping of housing data.

The integration of SeleniumBase and BeautifulSoup provides a robust solution for web scraping, capable of handling dynamic content and extracting relevant information accurately. The custom YAML storage solution

with TinyDB ensures that the data is easily accessible and modifiable, promoting flexibility in data management.

Geopy's geolocation capabilities further enhance the system by converting address data into precise geographic coordinates, allowing for advanced spatial analysis. This integration of various tools and libraries into a cohesive system demonstrates the power of Python for developing sophisticated data extraction and analysis solutions.

V. Conclusion

This paper demonstrates a robust approach to automating the extraction, storage, and analysis of housing data. The system's modular design, leveraging powerful Python libraries, ensures flexibility, scalability, and ease of maintenance.

Future work includes extending the system to multiple sources to aggregate more comprehensive datasets. Enhancements can be made to the data analysis capabilities by integrating machine learning models to predict housing trends and prices based on historical data and other features. Additionally, expanding the geolocation analysis to include proximity to amenities and public transportation can provide deeper insights into housing data.

Another potential improvement is optimizing the web scraping process to handle a wider variety of websites with different structures and layouts. This can be achieved by implementing more sophisticated parsing techniques and incorporating AI-based models to identify and extract relevant information.

In summary, the system provides a valuable tool for researchers, real estate professionals, and policymakers to analyze housing data efficiently. By automating the data extraction process and integrating spatial analysis capabilities, the system offers a comprehensive solution for understanding and leveraging housing market trends.

References

[1] M. Mintz, “SeleniumBase: Python Web Automation and E2E testing,” SeleniumBase. https://seleniumbase.io/

[2] L. Richardson, “Beautiful Soup: We called him Tortoise because he taught us.” https://www.crummy.com/software/BeautifulSoup/

[3] “Welcome to TinyDB! — TinyDB 4.8.0 documentation.” https://tinydb.readthedocs.io/

[4] “Welcome to GeoPy’s documentation! — GeoPy 2.4.1 documentation.” https://geopy.readthedocs.io/