Building Business Intelligence Data Extractor using NLP and Python

Tamilselvan Arjunan,
Assistant Manager, Ernst and Young strategy,
Data Science and Analytics

Abstract:- The goal of the Business Intelligence data extractor (BID- Extractor) tool is to offer high-quality, usable data that is freely available to the public. To assist companies across all industries in achieving their objectives, we prefer to use cutting-edge, business-focused web scraping solutions. The World wide web contains all kinds of information of different origins; some of those are social, financial, security, and academic. Most people access information through the internet for educational purposes. Information on the web is available in different formats and through different access interfaces. Therefore, indexing or semantic processing of the data through websites could be cumbersome. Web Scraping/Data extracting is the technique that aims to address this issue. Web scraping is used to transform unstructured data on the web into structured data that can be stored and analyzed in a central local database or spreadsheet. There are various web scraping techniques including Traditional copy-and-paste, Text capturing and regular expression matching, HTTP programming, HTML parsing, DOM parsing, Vertical aggregation platforms, Semantic annotation recognition, and Computer vision webpageanalyzers. Traditional copy and paste is the basic and tiresome web scraping technique where people need to scrap lots of datasets. Web scraping software is the easiest scraping technique since all the other techniques except traditional copy and pastes require some form of technical expertise. Even though there are many webs scraping software available today, most of them are designedto serve one specific purpose. Businesses cannot decide using the data. This research focused on building web scraping software using Python and NLP. Convert the unstructured data to structured data using NLP. We can also train the NLP NER model. The study's findings provide a way to effectively gauge business impact.

The solution has a greater impact when applied to:
- **Analyzing companies' fundamentals**
- **Analyzing better deal opportunities.**

Keywords:- *Web Scraping, Information Extraction.*

I. INTRODUCTION

Business Intelligence data extractor can be used by many of the world's leading industries to convert millions of web pages into meaningful information daily.To effectively gauge the impact on business, this solution might be made available as a service.

The following factors increase the impact of the solution:
- Fundamental analysis of companies
- Analyse prospects for better deals.

Data as a Service (DaaS) enables intelligent decision-making by providing high-quality structured data to improve business outcomes, acquire useful insight, and boost business outcomes. Any research, whether it be academic, marketing-related, or scientific.

People may desire to gather and examine the information from several websites. the various websites that belong information shown according to the particular category are varied formats. You might not be able to compete even with one website to view all information at once. Data spans are possible. spans several pages and under different topics.The only available method is manually copying the website's data into a local file on your computer. This is an extremely time-consuming and laborious task.

II. OVERVIEW OF WEB DATA EXTRACTION

Web data extraction is a fantastic method for removing and converting unstructured data from websites. The information into structured information that may be stored anddatabase-based analysis Web scraping also goes by the namesWeb harvesting, web data extraction, and web data scrapingor scratching the screen. Data collected by web scraping is called mining. The process of web scraping is intended to:information from websites is extracted and transformed intoa logical framework, such as databases and spreadsheetsor a CSV (comma-separated values) file.

A. Challenges

Targeting websites, such as the "top 100 search results for this phrase" or "these 3 e-commerce websites for this product category," is the first step in web scraping. On the surface, this may seem simple, but the next step requires finding precise URLs that match these targets, which is difficult for web scraping. To create the target URLs for the required pages, a web scraper must locate the source URL. Broken links and websites with irrelevant information cause the algorithm to waste time and data storage while creating

thousands of URLs for content that has no commercial value to the consumer.

To avoid having their services interrupted by heavy traffic, websites may try to prevent web scrapers. They accomplish this by "fingerprinting" the scraper in order to identify its origin and behavior. Examples of this include determining whether the same IP address is repeatedly attempting to scrape the same website, the scraper's device and operating system, and the speed at which requests are sent. According to a study, these fingerprints can be followed by websites once they have been recognized for an average of 54 days. This necessitates the usage of unique origins in each online scraping request and the requirement for web scrapers to anonymize themselves by altering their behavior to that of human users while scraping a website.

B. Solution

Web scraping is made easier by AI in two ways:

Algorithms for classifying data: Algorithms that have been trained on large data sets obtained via web scraping can recognize and categorize inactive URLs. This enables web scraping algorithms to focus their efforts on only a small fraction of potentially useful websites.

Algorithms for natural language processing: A recent study recommends enhancing web scraping algorithms to use natural language processing to scan the scraped data and determine the content's relevance. In this method, the effort required for data processing and storage is optimized because data that is below the relevancy level would not be saved at all.

Dynamic proxies, which require the web scraper to dynamically alter their IP address with each web scraping request, are a frequent solution to this problem. Other factors do, however, still aid websites in identifying automated web scrapers. Dynamic proxy technology is supported by AI solutions that optimize the other parameters. Web scrapers can use this training data to make sure the new parameters they employ are considerably different from the fingerprints they generated in thepast as each attempt at web scraping generates a fingerprint on the scraper end.

AI techniques can produce adaptive parsing models that gain knowledge via practice. Parsing models can learn how to effectively classify distinct sections of the scraped data and weed out unneeded pieces by utilizing parsed data as a training set. Some of these features, despite having separate website structures, might also be present on related websites. For instance, a data parsing algorithm may identify the approximate location of a product's image and details and use this as a proxy to identify where to look for the necessary data in a different dataset because many e-commerce websites have similar layouts to display the product image and details, such as price.

We will use SpaCy to train custom NER model. SpaCy is an open-source software library for advanced natural language processing, written in the Programming languages Python and Cython were used to create the open-source software library known as SpaCy, which is used for sophisticated natural language processing. To train our custom-named entity recognition model, we'll need some relevant text data with the proper annotations. We will use open-source US data to train the NER model.

In contrast to NLTK, which is frequently used for research and education, spaCy concentrates on offering software for use in actual production. By combining statistical models trained by well-known machine learning libraries like Tensor Flow, PyTorch, or MXNet through its machine learning library Thinc, spaCy now supports deep learning workflows as of version 1.0.
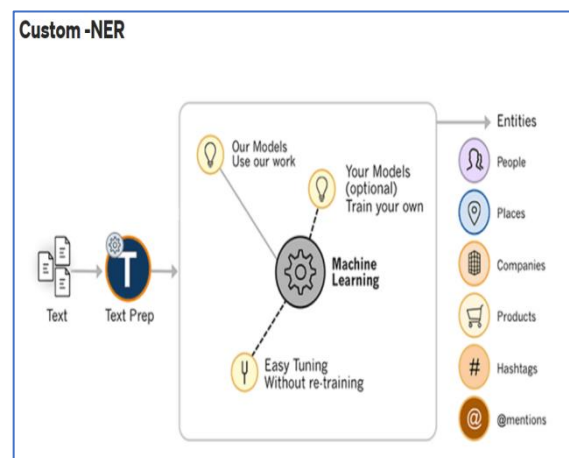

Fig. 1: Custom NER

Now, the major part is to create custom entity data for the input text where the named entity is to be identified by the model during the testing period.

At its core, all entity recognition systems have two steps:

Detecting the entities in text Categorizing the entities into named classes. In the first step, the NER detects the location of the token or series of tokens that form an entity. Inside-outside-beginning chunking is a common method for finding the starting and ending indices of entities. The second step involves the creation of entity categories. These categories change depending on the use case, but here are some of the most common entities classes:

- Person
- Organization
- Location
- Time
- Measurements or Quantities
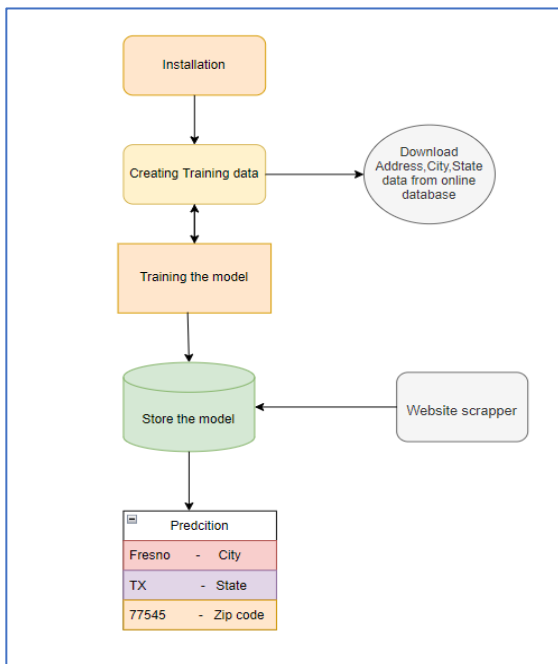- String patterns like email addresses, phone numbers, or IP addresses

Fig. 2: NLP NER Model building

Web scraping's significance in machine learning

Web scraping in machine learning is primarily focused on the fundamental issue of obtaining high-quality data.

Although the internal data gathered on routine business operations can offer insightful information, such data is insufficient. Therefore, even though it is a more difficult process, getting information from outside sources is crucial. When scraping, accuracy and poor data quality become major issues. As a result, every scraping project must always include a final clean-up process, although this will be covered in more detail later in this guide.

Fig. 3: Technical Architecture and Workflow

Below are the steps to be performed to get this tool running
- Select type to web scrapping – Text, Geo coordinates from maps or Images.
- Input the web URL from where data is to be extracted.
- Tool will navigate to the web URL and display the web page in the display panel
- User selects elements/links from where data is to be extracted
- Select if data is to be extracted from multiple pages.
- Run the tool and extract data

The following example collects historical stock prices using web scraping. Data points, such as daily opening, daily highest, daily lowest, and daily closing, will be collected as well.Thankfully, numerous websites provide such data, and it's usually, conveniently, presented in a table. Typically, you'll see the HTML code that renders these tables, such as the following image.

Dynamic Fingerprinting Powered by AI

How might AI- and ML-based anti-bot algorithms be best defeated? developing a crawling method that uses AI and ML. Finding reliable data is not difficult because the indicators of success and failure are clear-cut.Anyone who has previously engaged in web scraping ought to already have a sizable collection of fingerprints that could be valued. These fingerprints might be tagged, saved in a database, and used as training data.

Testing and validation, however, will be slightly more challenging. Some fingerprints may experience blocks more frequently than others because not all fingerprints are created equal. The AI will be significantly improved over time by gathering information on success rates per fingerprint and developing a feedback loop.
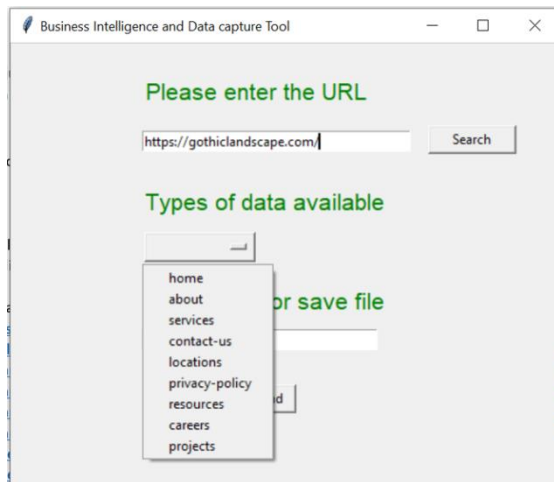
III. DESIGN OF SOFTWARE:

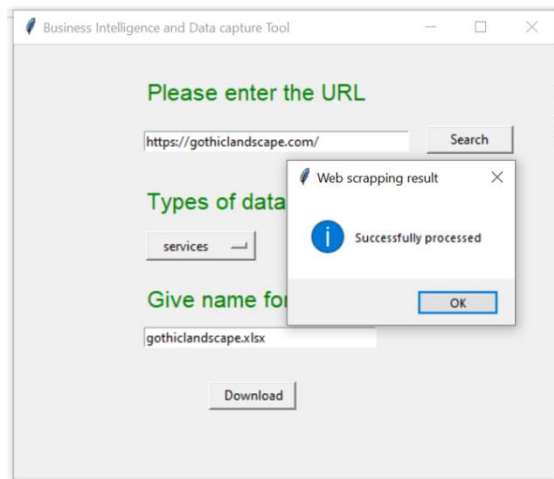Enter the URL and click on Search

Step 1:
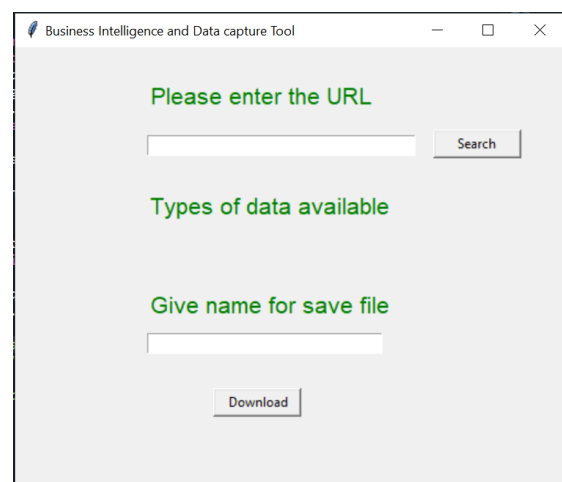
Select the type of data to be extracted

Step 2:

Click on download button

Step 3:

Output in excel file:

Fig. 4: Service data extraction

Fig. 5: Location data extraction:

IV. CONCLUSION

At some time soon, applying AI and machine learning to unstructured data will become inevitable. This business intelligence data extraction can help to create a financial news sentiment analysis to assess the effect on market value and other drivers that aid in strategic planning and assist management in identifying key strategic levers.

Building AI and machine learning models could appear to be a difficult undertaking to some people. Web crawling, however, is a game with a lot of moving components. It's not necessary to develop a single, all-encompassing ML model that could perform every task. Attend to the lesser chores first (such as dynamic user agent creation). Small ML-based models will eventually allow you to construct the whole web crawling system.

It can also help business to create an intelligent search engine to gain visibility into multiple competitors' products, services offered and their presence in different regions based on up-to-date and comprehensive data for making better deals and to get competitive edge.

REFERENCES

[1.] Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., &Preneel, B. (2013). Fpdetective: Dusting the web for fingerprinters. In Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. New York: ACM.

[2.] Bar-Ilan, J. (2001). Data collection methods on the web for infometric purposes – A review and analysis. Scientometrics, 50(1), 7–32. Butler, J. (2007).

[3.] Visual web page analytics. Google Patents.

[4.] Doran, D., & Gokhale, S. S. (2011). Web robot detection techniques: Overview and limitations. Data Mining and Knowledge Discovery, 22(1), 183–210.

[5.] Yi, J., Nasukawa, T., Bunescu, R., &Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, IEEE. Melbourne, Florida, USA.

Biography of Author

Tamilselvan Arjunan is working as an Assistant manager at Ernst and Young strategy. He has a total of 7 years of hands-on experience in Machine learning, Data Science and Python. He has built many AI-based products for clients. He is certified in Data Science and Python. Hecompleteda bachelor's degree in mechanical engineeringfrom Anna University.