International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-7, Issue-8, 2020, Regular Edition E-ISSN: 2349-7084

Developing Framework of Web Scraper for Agriculture Data using Client Server Module

[1]Prateeksha Nashipudi, [2]Dr. S. B. Kulkarni

[1]PG Scholar, SDM College of Engineering and Technology, Dharwad, Karnataka, India.
[2]Associate Professor SDM College of Engineering and Technology, Dharwad, Karnataka, India.
E-mail: [1]prateekshanashipudi96@gmail.com, [2]sbkulkarni_in@yahoo.com

Available online at: http://www.ijcert.org

Received: 30/07/2020 Revised: 06/08/2020 Accepted: 10/08/2020 Published: 01/09/2020

Abstract:- Searching for relevant information becomes very difficult and sometimes we don't find the exact information what we are actually seeking, so it results in time-consuming and repeating the same web page without knowingly. A system which will know our needs, requirements, preferences and patterns. This will retrieve the correct information and helps in fast processing. In this work, it is proposed that the personalized search engine for information retrieval system using the client-server module for user preferred information through intelligent search and storing the searched result in a database for further accessing of information is implemented. For information retrieval, a framework known as scrappy is used for retrieving all the user needed information by specifying the URL of that data. The fetched information is stored in the database. It helps in offline browsing, full-text search in the database and fast response and no repeating of web pages.

Keywords: web scraping, crawling, user agent, scrapy framework and web spider.

1. Introduction

As the rapid development of information grows exponentially on internet, enormous amount of data will be generated into the web just by taking the pictures, videos and posting it on face book, twitter, YouTube blogs etc; it is also an advantage to write anywhere and anytime and posting it. There are more than 16 million host computer on internet and more than billion web pages and huge amount of information is dumped into the World Wide Web on daily basis. Nowadays each and every person uses the internet for searching the information and it has become the most important application by using search engine. All the search engines like, Google, Yahoo, Bing and many more search engines gives the information to internet users. But these search engines are not personalized search engines, and these do not know the purpose of user needs, interest and preferences. So it becomes difficult to get the desired result for users, what exactly the user is seeking. It results into time consuming, less efficient, repeating the same web pages and links. The search engines searching mode is based on key word index. So it require correct word with exact meaning to be searched, if the word is inappropriate, then it will give all the information about it, which is out of scope for the user.

Now each and every person is using the internet and it has become open and distributed circumstances depends on network bandwidth. So search engines may not provide high precision, recall and searching speed. So this becomes difficult to search information and get accurate results.

The paper is organized as follows. Section "Literature Survey" discusses known efficient multi-agent infrastructure and their components used to develop information retrieval system. Section "Methodology" describes the architecture of the system, how web scraping and user agent is implemented and deployed. Section "Experimental observations" presents the results and outcome of the system. Finally, the evaluation of the web

© 2020, IJCERT All Rights Reserved https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0801

scraping architecture and contribution of the paper are presented in conclusions.

2. Literature Review

There are lots of surveys conducted to on the basis of user personalized information retrieval system. And some more relevant systems and architectures are discussed below.

(A) Multi-agent Architecture for User Adaptive Information Retrieval Systems is based on the implementation of software agents for information retrieval system based on user oriented model. [2] The existing system lacks the adaption to user requirements, so by taking user queries, history, preferences and account profile. The adaptive user preferred system is been implemented. Author uses JADE tool to implement software agents for communication purpose.

(B) Intelligent Agents for Cooperative Designs in Individual Information Retrieval, the author presents the drawback of traditional search engine and Implements a model for cooperative design in individual information retrieval based on intelligent agent's integration search engine. Author also presents the extra feature addressed in this paper, [8] offline browsing, intelligent search, personal information retrieval and full text search in database. Its client server based three layer architecture modules.

(C) A Model Based on Three-Layer Agent of Personalized Information Retrieval Systems, it is based on the three layer agent architecture of personalized information retrieval system. Author specifies the characteristics of intelligent agent in this paper, some of the features of intelligent agent are[9] Representation, filtration, management, discovery, intelligence, independence, flexibility, navigation and solution

3. Methodology

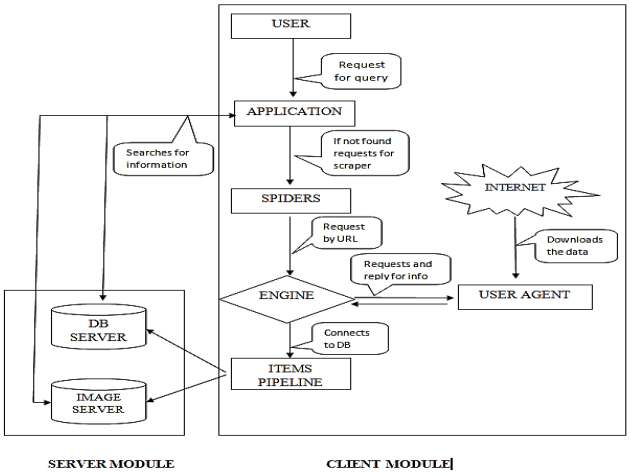


Fig 1. Web scraping and database model

Figure 1 consists of three modules, and they are: Web Scraping, Database server and image server, Application program and user. First the user needed information his/her pattern, preferences and history is collected and that specified information is extracted from the internet by specifying their website URL through scrappy framework, it will create a bot and this helps to build spiders for each URL which needed to extract the information from the website. The information which we need will be taken by specifying the XPath from the webpage. The engine will request for internet to download the specific web page of URL this requesting process is taken by the user agent from the browser and sends to the server, the requested data is processed and it will download those data and send it back to the user agent, this will carry the processed data to the scrapy engine the items pipeline is used to connect between engine and database. Information is extracted and stored in the database in the form of tables.

Application program contains searching option for the user, when the user searches for the query, application will check in the database, whether the specified information is present in it or not. If there is information then it will be displayed on the application otherwise again it searches the information through internet and scrapes the data and dumps into the database. All the information present in the database is classified as database server for filtering and sorting of only text information and image server for classifying the image data. Database and image clusters will act as server side information processing, and application will act as client side for requesting of information.

Scrapy comes with its own mechanism for extracting data. They're called selectors because they "select" certain parts of the HTML document specified either by XPath or CSS expressions. XPath is a language for

© 2020, IJCERT All Rights Reserved https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0801

selecting nodes in XML documents, which can also be used with HTML. CSS is a language for applying styles to HTML documents. It defines selectors to associate those styles with specific HTML elements integration with Scrapy Response objects.

Response objects expose a Selector instance on .selector attribute:

```
response.selector.xpath('//span/text()').get()
```

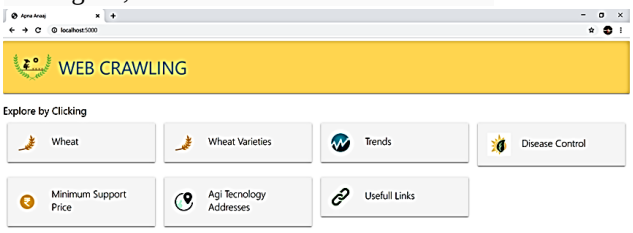
Scrapy selectors are instances of Selector class constructed by passing either TextResponse object or markup as an unicode string (in text argument). Usually there is no need to construct Scrapy selectors manually: response object is available in Spider callbacks, so in most cases it is more convenient to use response.css() and response.xpath() shortcuts. By using response.selector or one of these shortcuts you can also ensure the response body is parsed only once. But if required, it is possible to use Selector directly. Constructing from text:

```
from scrapy.selector import Selector
from scrapy.http import HtmlResponse
 response = HtmlResponse(url='http://example.com',
body=body)
Selector(response=response).xpath('//span/text()').get()
```

4. Experimental Observations

In this proposed work, collected only wheat information related to India as input data; this specific data is collected through user searched history as URLs, and scraped the relevant information according to user needs.

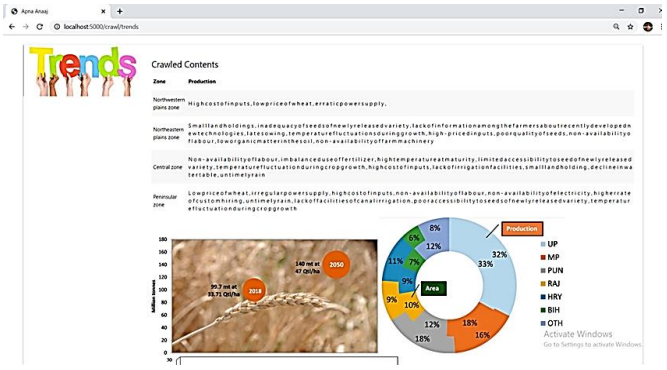
In Fig.2 its user interface personalized search engine which runs on the local host and resides on the client side. User can search the information what they are actually seeking for, it is based on wheat data in India.



[Input: Wheat Data, Country: INDIA]

Fig 2. User interface Application of personalised search engine

In Fig 3. It's the scraped data about the trends and prospects of wheat information, the scraped data will be stored in database in server side. When client request for information it will search for data and reply with relevant result. If result is not found in database, the application will scrape the information through internet and displays the result.



[Input: Wheat Data, Country: INDIA]

Fig 3 Scraped data

Table 1. Comparison of personalized search engine and with other search engine

	Personalised search engine	Other search engines
1	Knows the user interest, preferences and needs	Don't know the user interest.
2	Offline browsing allowed.	Offline browsing is not possible.
3	User query is searched easily as it is based on user interest	Needs specific query for searching otherwise it will display irrelevant information
4	No repeating of searched WebPages.	Repeating of searched WebPages.
5	High efficient and precision, searching time is minimised.	It requires more time to search

© 2020, IJCERT All Rights Reserved

Table 2. For wheat data following results are obtained:-

	Criteria Proposed Work	Other Search Engine
No of web pages scraped	300	0
No of repeated web pages	0	Possibility of repeating
Search time	3000 pages/min	Takes longer time than scrapy

By default, Scrapy uses a LIFO queue for storing pending requests, which basically means that it crawls in DFO order. This order is more convenient in most cases. While pending requests are below the configured values of CONCURRENT_REQUESTS, CONCURRENT_REQUESTS_PER_DOMAIN or CONCURRENT_REQUESTS_PER_IP, those requests are sent concurrently. As a result, the first few requests of a crawl rarely follow the desired order. Lowering those settings to 1 enforces the desired order, but it significantly slows down the crawl as a whole.

The throughput of the system to depend on the average time that it takes to download a page, which includes remote servers component and our systems latencies

$$t_{download} = t_{response} + t_{overhead}$$

The lag between the time to get a Response and the time its Items get out on the other end of the pipeline, as well as the time until we get the first responses and some inferior performance while caches are cold. Overall, if it is need to complete a job of N Requests and the Spider is properly tuned and it should be able to complete it in:

$$t_{job} = \frac{N.(t_{response}+t_{overhead})}{CONCURRENT\ REQUESTS} + t_{start/stop}$$

5. Conclusion

In this work, the proposed generalized architecture supports scrapy framework for information retrieval system based on URL provided by the user. The web scraper is developed for fetching of data by developing different spiders for the each URL, and user agent will carry the URL from client browser to server for requesting of information to be extracted and the fetched data is stored in the database. Its personalized search engine provides offline browsing, with full text search in the database and has high precision and recall. The scrapy only visits the specified URLs but other search engine visits all img file, css file to render page, so the searching time is more compared to normal search engines. Scrapy has scalability and speed with 1000 pages per second on commodity cloud based virtual servers. There is a limit of 2 MB of per request and requests are accepted until 60 seconds. Scrapy can crawl, download pages and even scrape content with high speed and efficiency with scrapping rate of 99.8%. The work can be further expanded by including multi agent software and communication between them, and expanding single client server module to distributed environment with multiple clients and server.

References

[1] Alvarado, Antonio Cuahtlapantzi, Eduardo Vázquez Santacruz, and Mariano Gamboa Zúñiga. "Construction of a basic intelligent agent." Intelligent Systems Conference (IntelliSys). IEEE, 2017.

[2] Pacheco-Reyes, Juan J., et al. "Multi-agent Architecture for User Adaptive Information Retrieval Systems." New Trends in Networking, Computing, Elearning, Systems Sciences, and Engineering. Springer, Cham, 2015.

[3] Shankhdhar, Gaurav Kant, and Manuj Darbari. "Building custom, adaptive and heterogeneous multi-agent systems for semantic information retrieval using organizational-multi-agent systems engineering, O-MaSE." 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA).

[4] Singh, Aarti, and Anu Sharma. "A framework for semantics and agent based personalized information retrieval in agriculture." 2nd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2015.

[5] Nunes, Ingrid, et al. "Dynamically Adapting BDI Agent Architectures based on High level User Specifications." Computer Science Department, King's College London(2010). [6] Luo, Junwei, and Xiao Xue. "Research on information retrieval system based on Semantic Web and multi-agent." International Conference on Intelligent Computing and Cognitive Informatics. IEEE, 2010.

[7] Czibula, Gabriela, et al. "IPA-An intelligent personal assistant agent for task performance support." 5th International Conference on Intelligent Computer Communication and Processing. IEEE, 2009.

© 2020, IJCERT All Rights Reserved https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0801

[8] Liu, Lizhen, Shujing Wang, and Hantao Song. "Intelligent agents for cooperative designs in individual information retrieval." 8th International Conference on Computer Supported Cooperative Work in Design. Vol. 2. IEEE, 2004.

[9] Xu, Yuanzhong. "A model based on three-layer agent of personalized information retrieval systems." International Conference on Image Analysis and Signal Processing. IEEE, 2011.

[10] THANGARAJ, MM. "Agent Based Personalized Semantc Web Information Retrieval System." International Journal of Advanced Computer Science and Applications 5.8 (2014)

© 2020, IJCERT All Rights Reserved https://doi.org/10.22362/ijcert/2020/v7/i08/v7i0801