**Designing Course Content Using a Customized Web Crawler Based On Search Keyword Topic**

Nilanjan Mhatre

*University Of North Carolina Charlotte*

*Charlotte, NC, United States*

[nmhatre@uncc.edu](mailto:nmhatre@uncc.edu)

Abstract

Online tutorials play a significant role in today’s world in digital transformation and research. They are easily accessible and readily available at moment’s need and contribute to faster learning and execution, by saving time. Although, in the present scenario, the knowledge base is expanding and updating so dynamically with everyday research and inventions, leading to varying content across all the numerous online tutorials available.

The availability of information on multiple sites may differ in content, that a person may not find the required information at some specific website/websites. The person is mostly forced to visit multiple websites for the same and sometimes combine the information from multiple tutorials to learn all the aspects of the desired topic or course.

A customized web crawler can be designed to fetch information from multiple online tutorials based on a user-defined keyword. A data mining model will extract relevant information from the collected information, along with key sub-topics, while discarding duplicate information. It will enable us to retrieve and display the complete aspects of the topic at a single place.

This project aims to use this concept to create a course on a topic, which would be a hybrid of multiple tutorials.

Introduction

A web crawler as defined in [5] is a bot program that parses the world wide web that will include all the websites that it can reach or are in its scope. As explained in [2], a web crawler will commence from a single web page. The “Web Scraping” process will download pages from the site and extract the content. It will then identify all the potential hyperlinks on that web page. These links could link to a different page on the same website or web application or the domain with a different URL or could direct the crawler to some other domain. The crawler then visits those links one by one and parses those pages for more content.

This content can be cleaned and then stored in a Database service that will be queried to get the content on demand quickly. Web crawling is time-consuming which is why this process is run beforehand so that the content is readily available.

This concept can be used to tackle multiple problems and can be incorporated into multiple applications like the ‘search engine’. A search engine will usually store keywords of the highest importance from the content received from the crawler. This is because storing the entire world wide web is impractical. The parsed web pages will be indexed to know which web pages contain what keywords.

This idea could be used in extracting only the required information from the world wide web. This project employs a “customized crawler” that scans multiple web pages containing tutorial content in its path and extracts its information. We will use “Academic Focused crawling” technique to accelerate the speed of our web crawler, by visiting and parsing Academic related documents, which are probably freely available. [8] gives the idea on extracting the data into a suitable format.

Once the information is acquired, as per [3], a Data mining algorithm will process the content. The most important technique in processing the data will be to eliminate duplicates along with other techniques like organizing the data. Once, we have the correctly structured data, we can use a software to display it on an interface along with customized features like searching, filtering, and navigation.

Each module will be deployed as a micro-service and will talk with each other through messaging services.

Academic-Focused Crawling

Focused crawling as developed in [6] is a system that targets a specific set of topics and web sites.

Our Focused crawler will consist of the components as follows.

1. Manager: This module creates multiple threads possibly on-demand to initiate crawling and ensure that no two crawler threads intersect or parse the same web page. The Manager defines the scope of crawling to optimize the process by configuring each thread. It manages the Database in terms of storing the content.
2. Crawler threads: The crawler threads are responsible to visit links iteratively and return the content to the manager.
3. Web Scraper: This is a part of crawler that scraps important information. It will identify spam data, useless links and advertises, and inform the parent crawler to ignore such pathways.

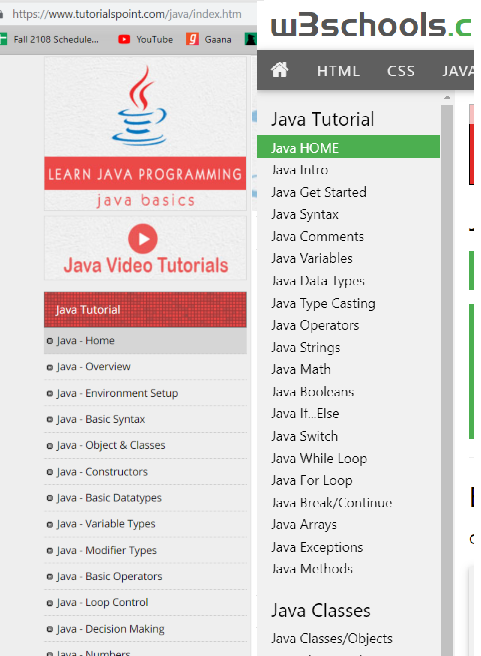


Figure 1: Websites being crawled

1. Data Pre-processor: The pre-processor is invoked by the Manager to identify important information from the content returned by the scrapper and structure it before dumping it into the database. The web page and its URL must be indexed.

The feasibility study conducted by [7] defines how to combine scraping and mining process.

We can optimize our crawling performance by initiating the crawlers from some specified popular academic websites.

Crawling will be carried in Priority Based Search, visiting websites first that is more relevant from the queue.

Revisit policy

Since the web has a dynamic nature and is likely to change, we will deploy a crawler which will crawl through indexed web pages in the database to keep track the of any updates taking place on the web page.

Data Mining

Once the Data is dumped into the database, it needs to be organized and structured. We will use concepts from [4] for mining. The most important issue will be the duplicate data. For example, we will get multiple definitions of a sub-topic from multiple domains.

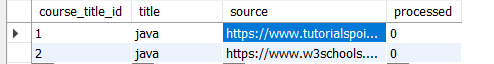


Figure 2: "Java" course crawled from multiple source websites

Text Related Preprocessing:

The following steps were taken to remove the noise created by the text in the dataset:

* Removing Unicode characters or words in other languages
* Stemming & lemmatization were also done in order to have effective modeling of content-based recommendation.
* Removal of English stop words, case-insensitivity and punctuation
* The extra white spaces were also been stripped.

Data Processing: -

1. Database Normalization: Normalizing the database is the most basic technique but proves to be very useful. But the normalization must be customized to work on the data acquired by a web crawler.
2. Choosing from duplicates: We will get multiple entries of sub-topics and multiple definitions of the sub-topics in the Database. We execute a strategy to choose the best of the duplicates to improve the quality of the content. The choice will be based on the score of each content.

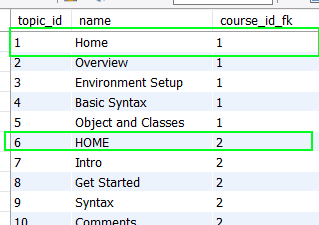


Figure 3: Duplicate topics received from multiple websites

1. Scoring the content: The score for content will be assigned by the quantity of the information, the source web page, and most importantly the number of unique keywords present in the content. The score represents how good is the data.

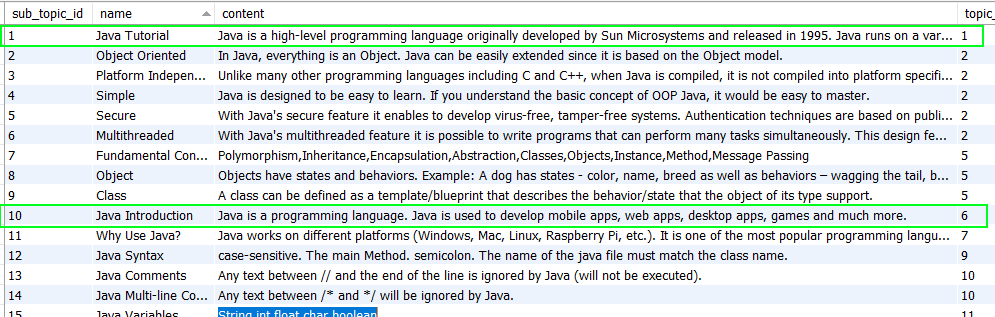


Figure 4: Combining topics and sub-topics

1. Calculating Frequent items for compressing data: After calculating the frequent item-sets and generating “Association rules”, we can filter out the important and relatable sub-topics. This is because we cannot include every topic we found. It also helps in ignoring a sub-topic found by one crawler, which is not a sub-topic at all.

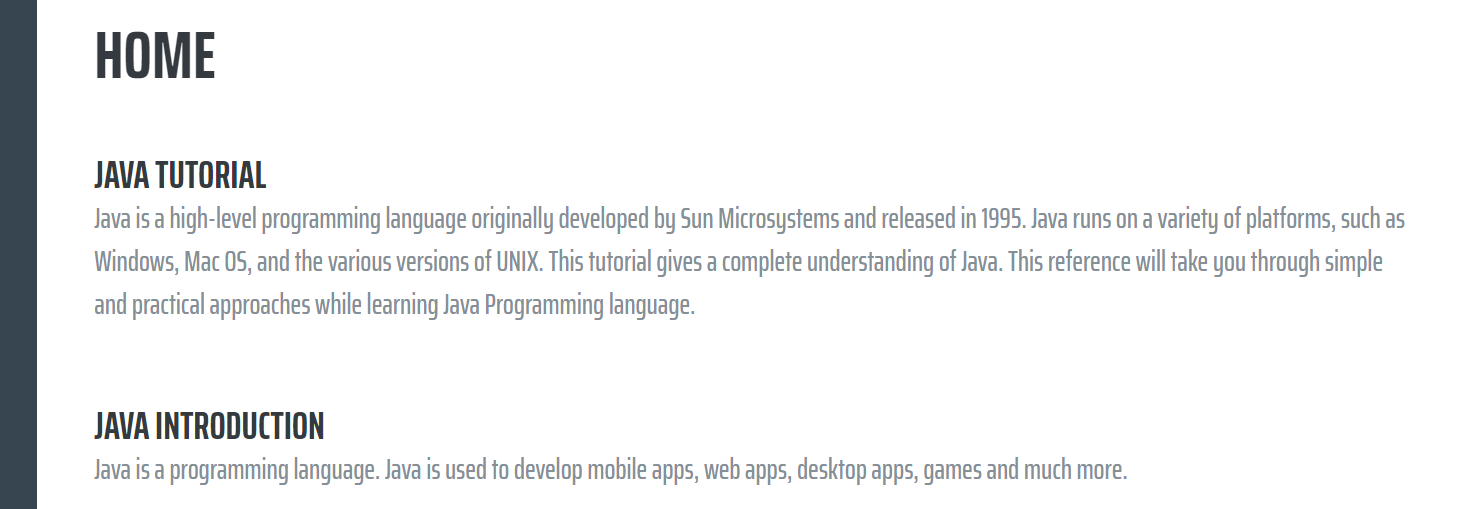


Figure 5: Combined information shown on the UI

JSoup

A library mentioned in [1], Jsoup is an open-source Java library that provides functions to parse an HTML DOM making it easy to manipulate the web scraping process.

* **Multiple Read Support** − Can read and parse HTML from URL or HTML string and can use CSS selectors too
* **DOM structure** – It reads the DOM as a structure and gives you selectors just like JavaScript
* **Prevents XSS attacks** – It uses a white list to clean user-submitted content, preventing spam data to be entered in the Database

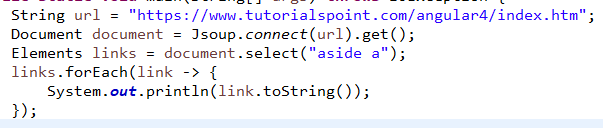


Figure 6: Web crawler code snippet

User Interaction

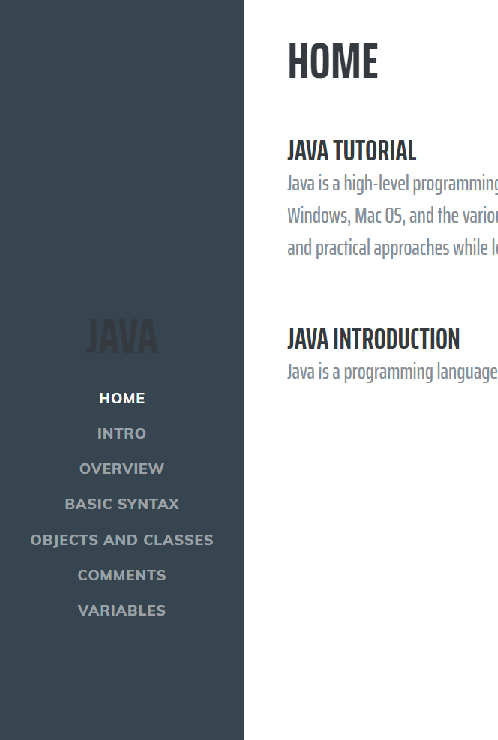


Figure 7: UI for displaying combined tutorial

The user interface will take an input keyword from the user and intimate the “Manager”. The “Manager” will search for possible topics from the Database and return the results. The user will select the topic and the application will generate the complete tutorial based on the information available such as the sub-topics, the definitions, and other content. This tutorial will contain the compacted information which will be the blend of all the information from multiple websites.

In case, we are not able to find any topic, the manager will deploy a job to initiate crawlers on high priority to crawl top web pages and generate a tutorial. It is possible that the jobs will be running in the background and updating the tutorial.

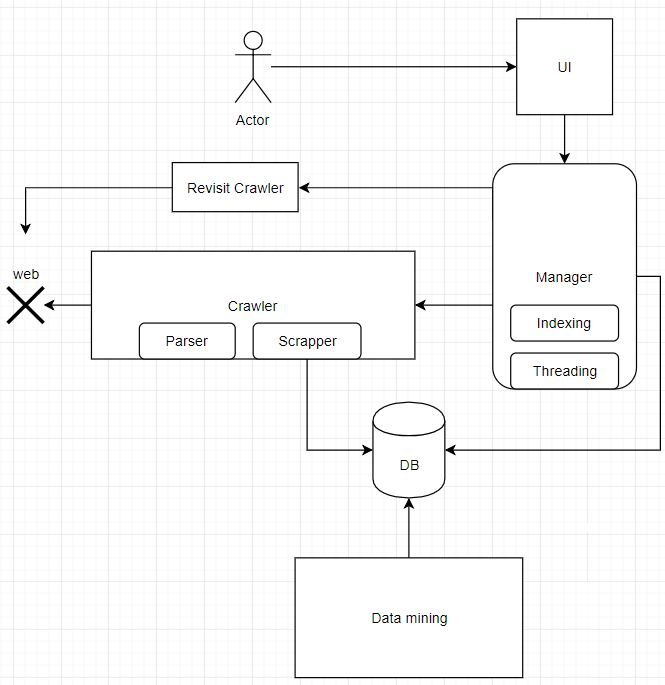


Figure 8: Microservice based system

Conclusion

This project aims at harnessing the ability of web crawling to gather the information from multiple web pages onto a single platform. This enables the user to view the required information in a compact manner. The filtering ability helps the user to customize the data mining functions.

The crawler will keep the data in the platform up to date.

References

[1] J. Hedley, “jsoup: Java HTML Parser”, <https://jsoup.org/>, https://www.tutorialspoint.com/jsoup/jsoup\_overview.htm, 2009

[2] C. Olston and M. Najork, “Web Crawling”, http://infolab.stanford.edu/~olston/publications/crawling\_survey.pdf, 2010

[3] S. Amudha and M. Phil “Web Crawler For Mining Web Data”, https://www.irjet.net/archives/V4/i2/IRJET-V4I225.pdf, *International Research Journal of Engineering and Technology*, 2017

[4] S. Pani, D. Mohapatra and B. Ratha, “Integration of Web mining and web crawler: Relevance and State of Art”, https://pdfs.semanticscholar.org/3a3c/18282c79106e8837a49496de9dbe1f138546.pdf, *International Journal on Computer Science and Engineering*, 2010

[5] PromptCloud, “Data Crawling & Real Time Bidding: Concept & Mechanism”, https://medium.com/@promptcloud/data-crawling-real-time-bidding-concept-mechanism-807ad8c0e0c7, 2015

[6] D. Bhatt, D. Vyas and S. Pandya, “Focused Web Crawler”, http://www.krishisanskriti.org/vol\_image/21Jul201512071432%20%20%20%20%20DAIWAT%20A%20%20VYAS%20%20%20%20%201-6.pdf, *Advances in Computer Science and Information Technology (ACSIT)*, 2015

[7] CRISP and National Training Fund, “Real-time labour market information on skill

requirements: feasibility study and working prototype”, https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/7/7e/FINAL\_REPORT.pdf, 2014

[8] M. Ahuja, J. Bal and Varnica, “Web Crawler: Extracting the Web Data”, https://www.ijcttjour nal.org/Volume13/number-3/IJCTT-V13P128.pdf, *International Journal of Computer Trends and Technology (IJCTT)*, 2014