AI - Based Solution for Web Crawling

Prashanth Kumar HM[1], Dr. Subramanya Bhat S[2]

[1]Research Scholar, College of Computer Science, Srinivas University, Mangalore, India
Email: *prashanth.hm02[at]gmail.com*

[2]Professor, College of Computer Science, Srinivas University, Mangalore, India
Email: *itsbhat[at]gmail.com*

Abstract: ***Web crawling, also known as web scraping or spidering, is the process of automatically gathering data from the internet. It involves using automated software tools using AI to visit websites, download data like web pages, pdf, videos, metadata, or images. Then store it in a structured format for later use. Web crawlers, also called spiders or bots, follow links from one webpage to another with AI validation. The information gathered by web crawlers can be used for a variety of purposes, including data mining, content aggregation, search engine indexing, market research or Plagiarism detection. Here our crawling is only for plagiarism detection, and our new AI based algorithms help us to do the fastest and most accurate data downloading.***

Keywords: Web Crawling, Structured Data, Link Validation, URL (Uniform Resource Locators), Artificial Intelligence

1. Introduction

Web crawling technology, when combined with AI (Artificial Intelligence), can make the process more efficient and effective. AI algorithms can be used to improve the accuracy of the data extracted by the web crawlers and to make better decisions about which pages to crawl next. Here's how web crawling technology works using AI:

Seed URLs: The web crawler starts with a list of seed URLs, which are the web pages that the crawler will visit first. These URLs can be chosen manually or generated programmatically based on specific criteria, such as a list of popular websites.

AI Decision - Making: With the use of AI, the crawler can make more informed decisions about which pages to visit next. For example, the AI algorithm can prioritize pages that are likely to have more relevant information based on the keywords, page title, or meta descriptions.

Fetching: The crawler visits the seed URLs and fetches the HTML content of the web page. It then extracts any hyperlinks found on the page and adds them to a list of URLs to visit later.

Data Extraction: Once the crawler has fetched the HTML content of a web page, AI algorithms can be used to extract relevant data from the page more accurately. For instance, the AI can identify and extract specific data points, such as product information or pricing data.

Natural Language Processing: NLP involves several techniques and approaches, including machine learning, deep learning, statistical modeling, and rule - based systems. These approaches are used to process text data, extract features, and train models that can perform specific tasks. NLP also relies on linguistic knowledge, such as grammar and syntax, to understand the structure of language and improve performance.

In web crawling technology is a process used to gather data from the internet domain. It involves automated software, called web crawlers or spiders, that systematically browse the web, following links between web pages, and collecting information along the way. [5]

Web crawling technology starts with a seed or base URL (Ex: https: //www.abc. com). A seed URL is a URL that the web crawler starts with, and it can be any URL on the web. The crawler then extracts all the hyperlinks URL on that page, which are the links to other web pages and ignores unwanted links. The crawler then visits each of these links, extracts more links, and continues this process until it has crawled a large portion of the web. [6]

The data collected by web crawlers can be used for a variety of purposes, such as search engine indexing, market research, and competitive intelligence. However, it is important to note that web crawling can be both legal and illegal, depending on the context and the methods used. Therefore, it is essential to respect the terms of service and copyright laws when using web crawling technology. It is an automated process, and it follows a set of rules and guidelines. These rules and guidelines are defined in the crawler's algorithm, which is a set of instructions that tells the crawler what to do and how to do it. The first step in web crawling technology is to identify the seed URL. The crawler then sends a request to the web server hosting the page, asking for permission to access the page. The web server will then respond with a status code, which indicates whether the page is available or not. If the page is available, the crawler will then start to download the page's HTML content. [7] The HTML content is the text that makes up the web page, and it contains all the links, images, and other elements on the page. The crawler will then analyze the HTML content, looking for links to other pages. The crawler then extracts the links from the HTML content and adds them to its list of URLs to crawl. The crawler will then follow each of these links, downloading the HTML content for each page and repeating the process of analyzing the HTML content and extracting links. Web crawling technology is a complex process, and there are many factors that can affect the crawler's behavior. These factors include the crawler's speed, the number of pages it can crawl, and the types of pages it can crawl.

2. Objectives

2.1 Link Validation

URL link validation refers to the process of checking whether a URL (Uniform Resource Locator) is valid or not. The validation process involves verifying if the URL is properly formatted and if it points to a web page or resource that exists and is accessible. Some of the common checks performed during URL link validation include:

Syntax check: This involves verifying if the URL follows the correct syntax and format, including the presence of the protocol (http, https, ftp, etc.), domain name, and file path.

DNS resolution: This involves checking if the domain name specified in the URL can be resolved to an IP address.

HTTP response code: This involves checking the HTTP response code returned by the server when attempting to access the resource specified in the URL. A response code of 200 indicates that the resource exists and is accessible, while other codes may indicate an error.

Content check: This involves checking if the content returned by the server matches the expected content based on the URL.

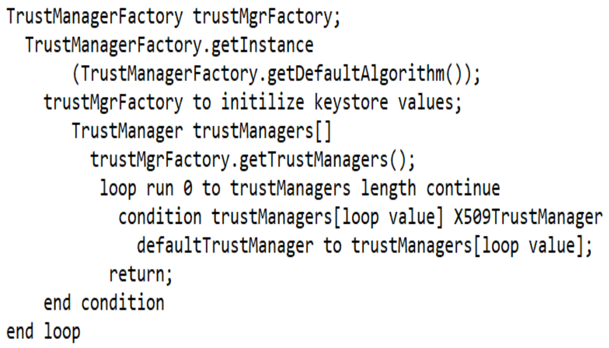
Here we implemented a validation algorithm using Open URL (parameters) libraries using Python. Python requests library, which can be used to send HTTP requests and handle responses.

2.2 URL Indexing

URL indexing refers to the process of adding web address or link to a search engine's database or index. Once a link is indexed, it may take some time before it appears in crawling results. This is because crawling engines use algorithms to determine the relevance and importance of web pages in relation to specific queries, and it can take time for these algorithms to process and analyse the newly indexed link. Here we can be categorized to two level one is newly generated URL and completed URL databases. Newly generated URL is fresh URL which is not visited in domain for crawling, and other end completed URL refers a crawled URL data system, which is visited URL, and which is generated to one or more URL from domain. Our indexing divided in to three level called domain level, naming level and sub address level.

2.3 URL Trust Manager

A URL Trust Manager is an algorithm that helps to protect URL from malicious connection by assessing the trustworthiness of URLs before allowing domain server to access them. The URL Trust Manager typically works by using a database of known malicious websites or by analysing various characteristics of the URL, such as the domain name, IP address, and SSL certificate. The following trust manager algorithm shows.



```
TrustManagerFactory trustMgrFactory;
  TrustManagerFactory.getInstance
       (TrustManagerFactory.getDefaultAlgorithm());
    trustMgrFactory to initilize keystore values;
       TrustManager trustManagers[]
         trustMgrFactory.getTrustManagers();
          loop run 0 to trustManagers length continue
            condition trustManagers[loop value] X509TrustManager
              defaultTrustManager to trustManagers[loop value];
           return;
    end condition
end loop
```

Here URL Trust Managers is also depending on machine learning advanced techniques to identify new and emerging threats.

2.4 SSL Context

An SSL context is a collection of cryptographic parameters that are used to establish secure communication channels over the internet. SSL (Secure Sockets Layer) is a cryptographic protocol used to encrypt data transmissions between a client and a server, and it has been succeeded by the newer TLS (Transport Layer Security) protocol. An SSL context typically includes information about the following:

Authentication: SSL contexts typically contain information about the digital certificates used to authenticate both the client and the server. This includes the public keys of the certificates as well as any associated metadata.

Encryption: SSL contexts specify the encryption algorithms used to secure the data being transmitted between the client and the server. These algorithms can vary depending on the level of security needed and the compatibility of the client and server.

Protocol Version: SSL contexts also specify the version of the SSL/TLS protocol being used. Different versions of the protocol have different security properties and capabilities.

Validation: SSL contexts contain information about how to validate digital certificates, such as which certificate authorities to trust and how to check certificate revocation status.

In our programming, SSL context is often used to refer to the collection of SSL/TLS parameters that are used in a particular SSL/TLS implementation. This includes libraries and tools that allow developers to create secure network connections between their applications and remote servers.

2.5 Protocol Monitoring

In our crawling technology mainly, we consider four types of protocol, they are 1. **HTTP** (hypertext transport protocol: It is a protocol used to access the data on the World Wide Web) 2. **IP** (Internet Protocol: is a protocol, or set of rules, for routing and addressing packets of data so that they can travel across networks) 3. **TCP** (Transmission Control Protocol: is a standard that defines how to establish and maintain a network conversation by which applications can exchange data) 4. **FTP** (File Transfer Protocol: is a network protocol for transmitting files between source to destination), here our AI protocol monitoring process maintain overall operational schemes, means from including URL sending to until receiving data or files over a network.

3. Process

Web crawling is the process of automatically collecting URL and data from website links on the internet. Here are some steps involved in web crawling:

Domin Pool: The first step in web crawling is to identify the website you want to crawl. This could be a single website or multiple websites. We called a one website in one domain here, we have to feed all our listed domain to domain pool, the thread and automatically allocate different domain for crawling.

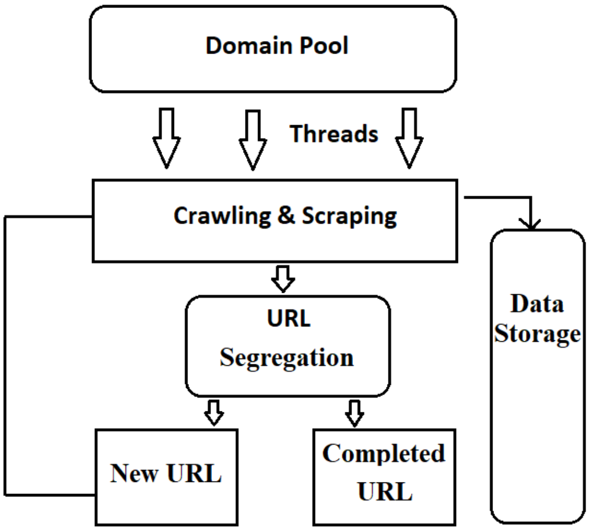


Figure: Schema model of complete process

Scope of the crawling: Once you have identified the domain to crawl, you need to define the scope of the crawl. This includes identifying the pages to crawl, the depth of the crawl (how many levels of links to follow), and any other parameters that are specific to our crawling needs.

Develop a crawling strategy: A crawling strategy we developed a set of rules and algorithms that dictate how the crawler should traverse all URL of the target website. Here we should start travel from seed URL, once crawl done by seed URL, from seed URL we get more than one sub - URL from same domain. Here we must maintain 2 sub data based one is new URL other is Completed URL.

Build the crawler: The next step is to build the crawler itself. This can be done using a variety of algorithms. Typically, the crawler will start by downloading the HTML content of a page and then parsing it to extract any relevant data to data base. The relevant data may be html text data, pdf data, images or videos.

URL Segregation: During collection of new URLS the algorithm can separate html pages (from same domain), images link, video links, pdf links. And during segregation our AI algorithm removed unwanted URL like, social media URL, adds URL, Redirection URL and other JUNK URL's.

Store the data: As our crawler collects data, it needs to be stored separately in a different database like text DB, image DB or Video DB. This could be a cloud - based storage solution for live indexing purposes.

In our overall implementation, web crawling is a complex process to developed by lot of planning, teamwork, programming, and technical expertise. However, it was a right tools and strategies, it can be a powerful way to gather valuable data from the web.

4. Implementation

4.1 Domain Collection

The domain pool contains millions of domains at academic level, this is an initial implementation part. Collection of domains is a risky job done by based group of AI algorithm and Proxy IP rotation, basically we must collect academic keywords and feed to google search by Google API automatically.

4.2 URL Threading

In our code can be possible to download millions of web pages within hour using multiple threads, basically in our calculation the average of web crawling is 3, 450 web pages crawl per minute using single thread, similarly in same duration 1200 pdf document, 2300 images and 340 videos.

4.3 Intermediate APIs

APIs that sit between the client crawling side and the domain server in a client - server architecture. The primary purpose of intermediate APIs is to act as a bridge between the crawling thread and URL server address, enabling communication between them and facilitating the download of data. Intermediate APIs perform various tasks, such as:

Data transformation: converting data from one format to another to make it compatible between the client and server.
Security: providing security features such as authentication, authorization, and encryption to ensure the safety of URL and data being transferred.

Caching: improving performance by temporarily storing data in memory so that it doesn't have to be fetched from the server every time it is needed.

Rate limiting: Controlling the rate at which the client can access the server to avoid overloading the server and ensure fair usage for all clients.

Intermediate APIs are commonly used in microservice architectures, where multiple services are deployed in a distributed environment, and in modern web and mobile applications to improve scalability and performance. In web scraping process it's very useful to carry user data or file to backend level, or processed data (file) to user level. [11]

4.4 User Interfaces

The goal of a UI in scraping data is to make the device's functions and capabilities easy to understand and use between link and domain server. In this process UI design takes into consideration the needs and preferences of the target links, as well as the devices and platforms the UI will run on. In our well - designed UI using AI, can enhance the accurate scraping and increase data transformation between two sides.

5. Issues and Handling

5.1 Robots. Txt

In robots. txt files indicate whether certain user agents can or cannot crawl parts of a website. These crawl instructions are specified by “disallowing” or “allowing” the behaviour of certain (or all) user agents. Robots. txt is a text file webmasters create to instruct web robots how to crawl pages on their website. The robots. txt file is part of the robot’s exclusion protocol (REP), a group of web standards that regulate how robots crawl the web, access and index content, and serve that content up to users.

5.2 Indexing pages

This indexing pages is a single web link, contains large data. Example some of the dictionary web pages having giga byte of the text data. While scraping of same page will take more time and creating huge traffic to domain. So overcome this issue we can fix timer to each URL and separate thread can be allocated to crawl by buffering concept.

5.3 URL Misguiding

Some of intelligence web developer aim to secure their web sites by avoiding crawling and traffic using some backend logics, one method commonly they used to misguide scrapper URL. When scrapper send to request to destination address, in the URL is redirected common junk page created by web developer, he is returning unwanted URL and data to scrapper.

5.4 URL Ambiguity

URL ambiguity refers to the situation where a single URL can refer to different web pages or resources depending on the context. These ambiguities can cause confusion for users and search engines and can also have implications for website security and accessibility. Web developers can mitigate URL ambiguity by using canonical URLs, implementing 301 redirects, and ensuring that URLs are consistent and meaningful.

6. Other Advantages

6.1 URL Traversal

URL traversal is an important process to avoiding duplication of scraper. Example, in the year of 2022 we have completely crawled domain, later we have expecting more new links released in 2023. In this case, if we do re scraping same domain, our AI ‘URL traversal’ identifies the only new available URL links to scrap from the year 2023 other past year link will be ignored.

6.2 Easy for indexing

It is the process of creating an index or catalog of the content of a collected of texts or other, to facilitate efficient searching, retrieval, and analysis. This process involves identifying the key concepts, terms, or entities that are important for describing the content of the texts, and creating an index that URL maps these concepts to the locations in the texts where they occur. This allows users to search for specific words or phrases and retrieve relevant texts quickly, rather than having to read through the entire collection. Text indexing is a fundamental technique in many information retrieval systems.

6.3 Data and URL binding

URL and data binding refers to the process of connecting a URL to a specific resource or functionality on a web server from the user side clicking link. When a user clicks a URL into plagiarised contents from the pdf or html page, the browser can be easily identifying the appropriate targeted resource in the same domain address.

7. Conclusion

The plagiarism checker works on different levels indexed of data, mainly those levels are live data, own database other is repository data. Here ‘DrillBit Plagiarism’ has its own database for downloading and storing all academic related data including journals, university repositories, MOOC repositories, e - learning, eBook etc. . . The crawling process is collecting billions of web data with short duration, here we have developed same to collect various academic level data to be stored and indexed in our own cloud database. This project implemented by artificial intelligence and some of natural language processing concept by expert development team.

References

[1] Russell, S., and Norvig, P.2002. “*Artifical Intelligence”: A Modern Approach. Prentice Hall*, 2nd edition.

[2] Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., & Preneel, B. (2013). “*Fpdetective: Dustingthe web for fingerprinters. In Proceedings of the 2013ACM SIGSAC conference on computer & communica - tions security. ”* New York: ACM

[3] Hirschey, J. K. (2014). “*Symbiotic relationships: Pragmaticacceptance of data scraping”.* Berkeley Technology LawJournal, 29, 897.

[4] M. Jazayeri. “*Some Trends in Web Application Development*”. In: Future of Software Engineering (FOSE ’07). May 2007, pp.199–213. doi: 10.1109/FOSE.2007.26.

[5] R. Diouf et al. “*Web Scraping: State - of - the - Art and Areas of Application*”. In: 2019 IEEE Inter. Conference on Big Data. Dec.2019; doi: 10.1109/BigData47090.2019.05594.

[6] Erdinç Uzun et al. "*EVALUATION OF HAP, ANGLESHARP AND HTMLDOCUMENT IN WEB CONTENT EXTRACTION*". In: Nov.2017.

[7] Y. Chu et al. "*Automatic data extraction of websites using data path matching and alignment*". In: 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC). Oct.2015, pp.60–64. doi: 10.1109/ICDIPC.2015.7323006.

[8] Krishna Kalyanathaya, Akila D., and Suseendran G. "A Fuzzy Approach to Approximate String Matching for Text Retrieval in NLP". In: Journal of Computational Information Systems 15 (May 2019), pp.26–32.

[9] *How to Scrape Data from PDF Files Using Python* reference web site https: //www.towardsdatascience. com

[10] D. Vernon, G. Metta, and G. Sandini, "*A survey of artificial cognitivesystems: Implications for the autonomous development of mental ca - pabilities in computational agents,* " IEEE Transactions on EvolutionaryComputation, vol.11, no.2, pp.151–180, 2007

[11] Almaqbali, I. S., Al Khufairi, F. M., Khan, M. S., Bhat, A. Z., Ahmed, I. (2019). Web Scrapping: Data Extraction from Websites. Journal of Student Research.

[12] Alrashed, T., Almahmoud, J., Zhang, A. X., Karger, D. R. (2020). ScrAPIr: *Making Web Data APIs Accessible to End Users. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (*pp.1–12). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3313831.3376691

[13] Chamoso, P., Bartolomé, Á., García - Retuerta, D., Prieto, J., De La Prieta, F. (2020). *Profile generation system using artificial intelligence for information recovery and analysis. Journal of Ambient Intelligence and Humanized Computing*, 11 (11), 4583 - 4592.

[14] Grasso, G., Furche, T., Schallhart, C. (2013). *Effective Web Scraping with OXPath. Proceedings of the 22nd International Conference on World Wide Web (*pp.23–26). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2487788.2487796.