Department of Electrical and Computer Engineering

North South University

CSE499B Senior Design Project

Bengali Website Scraper

Submitted By:

Name: S.M. Istiak Ahmed
ID # 1721365042

Name: Efat Hossain Emon
ID # 1711911042

Faculty Advisor:

Mohammad Ashrafuzzaman Khan
Section: 07

LETTER OF TRANSMITTAL

May, 2021

To

Dr. Mohammad Rezaul Bari

Associate Professor and Chairman,

Department of Electrical and Computer Engineering,

North South University, Dhaka.

Subject: Submission of Capstone Project on “Bengali Website Scraper”.

Dear Sir,

With due respect, we would like to submit our Capstone Project Report on “Bengali Website Scraper” as a part of our BSc program. The report deals with an optimized algorithm to scrape text data from most Bengali websites and allows for direct storage of scraped date into a categorized database. It may also assist the users in decision making regarding a topic by collecting the relevant data very easily and quickly. We tried our level best to make the report meaningful and informative.

The Capstone project was very much valuable to us as it helped us to gain experience from practical field. It was a great learning experience for us. We tried to the maximum competence to meet all the dimensions required from this report.

We will be highly obliged if you are kind enough to receive this report and provide your valuable judgment. It would be our immense pleasure if you find this report useful and informative to have an apparent perspective on the issue.

Sincerely Yours,

....

S.M. Istiak Ahmed

Department of ECE

North South University, Bangladesh

....

Efat Hossain Emon

Department of ECE

North South University, Bangladesh

APPROVAL

The capstone project entitled “Bengali Website Scraper” by S.M. Istiak Ahmed (ID # 1721365042) and Efat Hossain Emon (ID # 1711911042), is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on September, 2021 and has been accepted as satisfactory.

Supervisor:

Dr. Mohammad Ashrafuzzaman Khan

Assistant Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Department Chair:

Dr. Mohammad Rezaul Bari

Associate Professor & Chairman

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

This is our truthful declaration that the “Bengali Website Scraper” project report we have prepared is not a copy of any other reports previously made by any other team. We also express our honest confirmation in support of the fact that the said report has neither been used before to fulfill any other course related purpose nor it will be submitted to any other team or authority in future.

....

S.M. Istiak Ahmed

Department of ECE

North South University, Bangladesh

....

Efat Hossain Emon

Department of ECE

North South University, Bangladesh

ACKNOWLEDGEMENT

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report. Next, we would like to thank our faculty, Dr. Mohammad Ashrafuzzaman Khan, who has been very helpful regarding this project and has helped us numerous times with guidance.

We would also like to give thanks to my family and friends for supporting us during this year.

ABSTRACT

The process of collecting data from the internet is known as web scraping. In a time when the internet is rich with data and there isn't enough time to look through it all, web scraping has become much more necessary and feasible to utilize in many applications. Beautiful Soup and the Python library requests are effective tools for the purpose. This project intends to extract text content from websites using web scraping to make it easier to detect a pattern in the information store or find relevant queries. The data is then stored into a database. As, User can get necessary information more easily and query more effectively. This project is focused on scraping Bengali web pages and attempting to determine the most efficient method of doing so.

Contents

CHAPTER 1: INTRODUCTION

The amount of data kept online has only grown over time. The use of the internet and cloud servers has increased dramatically in recent years. As a result, searching for valuable data from available sources has become extremely challenging. Data is an essential part of any research, either it can be academic, marketing or scientific (SysNucleus, n.d.). People might want to collect and analyse data from multiple websites. The different websites which belongs to the specific category displays information in different formats. Even with a single website you may not be able to see all the data at once. The data may be spanned across multiple pages under various sections. Most websites do not allow to save a copy of the data, displayed in their web sites to your local storage (Penman et al., 2009). The only option is to manually copy and paste the data shown by the website to a local file in your computer. This is a very tedious job which can take lot of time. Web Scraping is the technique which people can extract data from multiple websites to a single spreadsheet or database so that it becomes easy to analyse or even visualize the data. (A Comparative Study on Web Scraping SCM de S Sirisuriya)

1.1 AIMS AND OBJECTIVES

This project sets out to conduct research in the area of web scraping and how it can be used as a tool for finding useful data and categorically store them. The main focus of this project is to scrape Bengali websites and save the scraped text data in database. The core objectives of this project are to:

- Develop a system that will use the most suitable algorithm to scrape the intended website
- Scrape only the specified types of data that the user intends to
- Categorize the scraped data and store them in a secure database
- Let the user search for specific tags in the scraped websites and also in the database

Furthermore, after the initial development of the project our aim and goal has been directed toward:

- Benchmarking various web scraping approaches to determine their accuracy and success rate
- Finding out the most optimal approach to scrape Bengali websites

CHAPTER 2: LITERATURE REVIEW

While working on this project, we studied various research papers related to our project topic and picked a few papers from there which were conducted on website scraping using different methods, libraries and scripts. We chose these particular papers because their working approach is closely related to our work. We also deduced some ideas from there for doing our work.

2.1 EXISTING LITERATURE EXPLANATION

Web scraping is a technique for automatically extracting data from a website. To develop customized scrapers, there are numerous frameworks and Application Programming Interfaces and also some configurable ready-to-use scraping tools are available. ‘‘Web scraping: Applications and tools,’’[8] gives idea about different types of web scraping techniques. Glez-Pea et al. [1] and Haddaway [2] provide detailed overviews of frameworks and tools for various extraction tasks.

The World Wide Web, currently present a huge amount of information in different formats and from different origins. Moreover, in many circumstances, this is the only source of information available to public. Nonetheless, extracting data from websites is tough, with one of the most difficult difficulties being automatically recognizing ‘‘the appearance of items of interest and their features on the web pages and saving them in a database in a uniform manner" [3] on websites.

An important decision when developing a web scraping tool is, therefore, whether to develop one’s own application (see, for example, [6]) or to use an existing tool. ‘Legality and Ethics of Web Scraping’[7] describe whether web scraping is legal or illegal and the Ethics of the web scraping.

2.2 RELATED WORKS

Some related works and their brief descriptions are given below:

1. Visual Web Ripper Visual: Web Ripper is one of the most advance web scraping software, created by Sequentum group in 2006 that provides functionality that allows you to scrape data from any websites like Business Directories, Simple Web Pages, Classified Sites, Forums and e-commerce site scraping (eBay, amazon, magento sites). Once data scraping finish, data can be exported to structured CSV, Excel, or XML format (List of Web Harvester, Data Scraper,Web Scraping Software and Tools)

2. Web Content Extractor: Web Content Extractor (WCE) is a simple user-oriented application developed by Newprosoft. It has good wizard that guide user to setup scraper. You can scrape data from website with few clicks and Web Content Extractor is excellent for putting data into different formats like Excel, text, HTML formats, Microsoft Access database, Structured Query Language(SQL) Script File, MySQL Script File, Extensible Markup Language (XML) file, HTTP submit form and Open Database Connectivity (ODBC) Data source. (“List of Web Harvester, Data Scraper, Web Scraping Software and Tools,” n.d.) (“Software for Web Scraping,” n.d.).

3. Mozanda Web Scraper: Mozanda Web Scraper is powerful web data extraction service. It can extract data from websites as well as PDFs. It has simple Point and selection interface so nontechnical can also make simple scrape. Mozenda runs your scraping project (agent) on their cloud environment which is the main difference of Mozanda from other scrapers. (“List of Web Harvester, Data Scraper, Web Scraping Software and Tools,” n.d.)

4. UiPath: Robotic Process Automation UiPath can automatically log in to a web site, extract data spanning multiple webpages, filter and transform it into the format of user choice, before integrating it into another application or web service. UiPath resembles a real browser with a real user, so it can extract data that most automation tools cannot even see (Savinkin, n.d.). No programming is needed to create intelligent web agents using its drag-and-drop graphical designer-but the .NET hacker inside you has complete control over the data (“List of Web Harvester, Data Scraper,Web Scraping Software and Tools,” n.d.).

5. Out Wit Hub: The OutWit Hub is a powerful Firefox extension Tool for everyone. The contents extracted from a web page are presented in an easy and visual way, without requiring any programming skills or advanced technical knowledge. Users can easily extract links, images, email addresses, data tables, etc. from series of pages without ever seeing the source code. Extracted data can be exported to CSV, HTML, Excel or SQL databases, while images and documents, are directly saved to your hard disk. The OutWit Hub is best to use for beginners in web scraping (“Software for Web Scraping,” n.d.).

CHAPTER 3: METHODLOGY

This chapter provides a chronological outline of the various elements of the work. It mostly explains the work's theories, methodology, and step-by-step procedure.

3.1 WORKFLOW

To solve this project, we followed certain steps. These steps can be easily explained using a flowchart:

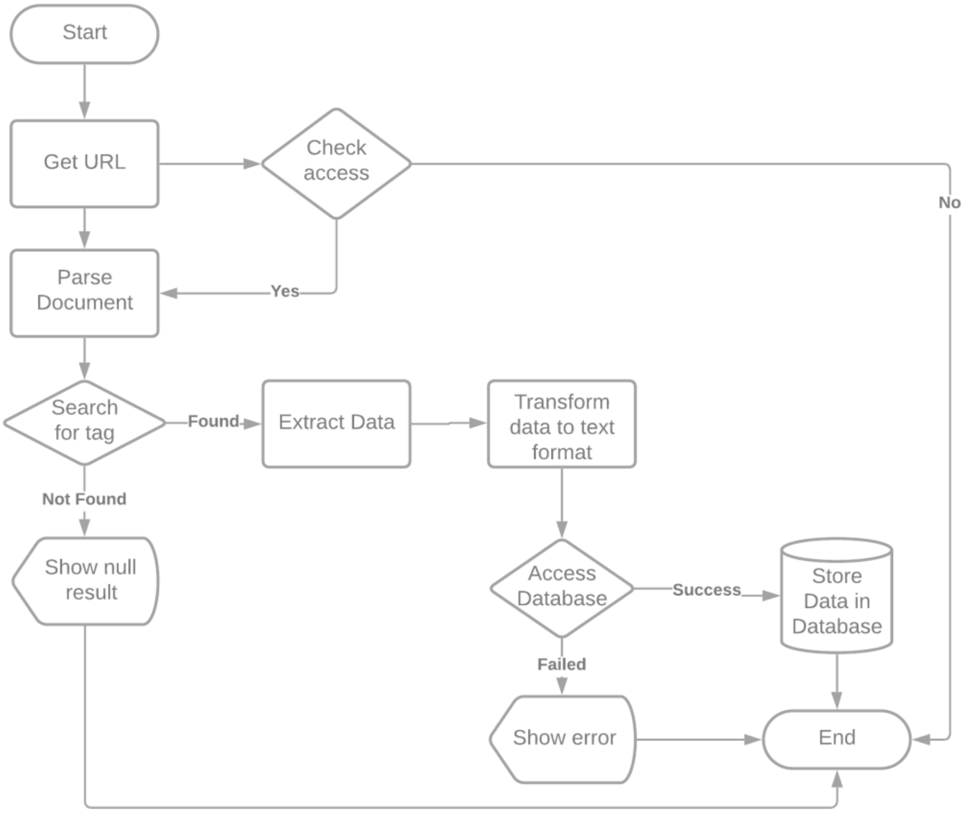


Figure: Flowchart of Bengali Web Scraper

3.2 DATASET

The data and the model are intimately connected. Although for this project, we did not require the typical datasets, we used around 100 Bengali websites and scraped their data.

3.3 DESIGN

To design a web scraper to serve our purposes, we will follow simple yet logical steps:

Step 1: Define our data requirements

Step 2: Conduct a legal review

Step 3: Evaluate the technical feasibility

Step 4: Architect a solution & estimate resources

We will be trying to clarify and investigate the feasibility of this web scraping project you should always be answer questions like:

- What data do we require?
- From which websites would we like to obtain this data?
- How often would we like to extract this data? How do you want to consume the data?
- How will you verify that the extracted data is accurate? i.e. matches exactly the data on the target websites?
- How would you like to interact with the solution? i.e would you just like to receive data at a predefined frequency, or would you like to have control over the entire web scraping infrastructure and the associated source code?

3.4 DIFFERENT APPROACHES

To fulfill the newfound focus of our project, we used 4 approaches in total to approximate how they deal with the problem of web-scraping and how their results vary. A very brief idea about the approaches are given below:

Mimicry Approach: This category of scraper works thanks to predefined customised rules. The location of the data to be collected from a web page is preconfigured in the scraper. This mechanism is applied on DOM selectors which are deduced from click based leaning. This strategy is relatively efficient thanks to its neatness, but is less adapted when it comes to process multiple heterogeneous websites. Furthermore, if the source website modifies its graphic design, the engine should be reprogrammed to how to find the needed information. Tools such as Import.io or Mozenda use this approach.

Differential Approach: This approach is based on the fact that two pages from the same website will only differ in content from the body of the page. According to this logic, the menu bars, the right or left columns, and the footers are supposed to be perfectly identical between two pages of the same website. The mechanism formerly consists in applying a masking algorithm that superimposes the two pages by removing only the differences.

Weight Measurement Approach: This approach calculates the total weight of words in each branch of the DOM tree and using deduction chooses the start and end node of the main content. It needs no training and is very adaptable to even changeable website designs. However, the results can be very noisy

.

Machine Learning Approach: In this approach, we train an algorithm on a large sample of manually analyzed web pages and try to make it learn the geographical location of the text block statistically. It takes a long time to teach but is quite flexible and the result gets better with sample size.

3.5 RELEVANT EXPLANATIONS

To understand how we developed this project, we must understand the basic components we used in this project. A short description of them are given below:

Beautiful Soup: Beautiful Soup is a Python library that extracts structured data from websites. It can parse HTML and XML files for details. It works as a helper module, interacting with HTML in a way that is close to and better than how you can communicate with a web page using other developer tools. It saves the programmer time since it has popular parsers like lxmal and htmal5lib.Beautiful soup's intelligence to convert incoming documents to Unicode and outgoing documents to UTF-8 is another important and useful function.In comparison to other parsing or scraping methods, it is also thought to be quicker.

Types of Parser and their Usage:

Parser	**Typical usage**	**Advantages**	**Disadvantages**
Python’s html.parser	**BeautifulSoup(markup,”html.parser")**	● Batteries included ● Decent speed ● Lenient (As of Python 2.7.3 and 3.2.)	● Not as fast as lxml, less lenient than html5lib.
lxml’s HTML parser	**BeautifulSoup(markup,"lxml")**	● Very fast ● Lenient	● External C dependency
lxml’s XML parser	**BeautifulSoup(markup, "lxml-xml") BeautifulSoup(markup, "xml")**	● Very fast ● The only currently supported XML parser	● External C dependency
html5lib	**BeautifulSoup(markup,"html5lib")**	● Extremely lenient ● Parses pages the same way a web browser does ● Creates valid HTML5	● Very slow ● External Python dependency

Spyder IDE: Spyder is a cross-platform IDE that is free and open-source. Python is used exclusively in the Python Spyder IDE. It was created by scientists for scientists, data analysts, and engineers only. It's also known as the Scientific Python Development IDE, and it comes with a long list of noteworthy features.

MongoDB: MongoDB is an open source, nonrelational database management system (DBMS) that processes and stores different types of data using versatile documents rather than tables and rows. It simplifies database management for developers and creates a highly scalable environment for cross-platform applications and services. We connected MongoDB API with the project to store the scraped data in database.

TensorFlow: Here TensorFlow is used to scrape data in machine learning approach. It is an open source artificial intelligence package that builds models using data flow graphs. Classification, perception, understanding, discovering, prediction, and creation are some of the most common uses for TensorFlow.

The Components of a Web Page

Knowing how web pages work is a crucial first step in every web scraping project, as we'll write code to get the data we want to scrape using the site's structure.

HTML: The HyperText Markup Language (HTML) is the language used to build web pages. HTML, on the other hand, is not a programming language like Python. It's a markup language that tells a browser how to view content in a particular way.

DOM Tree: The Document Object Model is a cross-platform and language-independent interface that treats an XML or HTML document as a tree structure wherein each node is an object representing a part of the document. The DOM represents a document with a logical tree.

Header: The header or banner is located on top of a website. It includes the logo of the company, the publisher or owner of the website. This automatically informs website visitors what the website is about. Websites that offer products and services usually have banners that feature their latest offers or even the current news about their company.

Navigation Bar: The navigation bar/menu tab allows the visitors to check other pages of the website. It appears in all pages within a website for more convenient navigation. Navigation bars are usually placed just below the banner/header for convenient access.

Footer: The footer is similar to sidebars. They have no particular design. However, they complement the overall design of the website. The footer is almost the same as the sidebar. The only difference is that they are located at the bottom part of a website

CHAPTER 4: RESULTS

After working on this project and modifying our code for the various problems we faced, we have managed to see substantial results. Although the project lacks a finishing touch as of now, the core objects are mostly being successfully completed by this project and we can call it a success.

The relevant results are shown and discussed below:

For the results, we have taken around 100 Bengali websites and tried scraping their data using the four approaches mentioned before. We calculated the success rate, total time taken and the accuracy of the scraped data and converted it into a bar chart.

The machine learning approach has faulty results as the sample size is still small and needs more sample and time to achieve optimal result.

To be noted that, Success rate refers to the percentage of successful perusal of data and clarity refers to how accurate the scraped data is compared to the data we wanted or if there was any unnecessary data scraped.

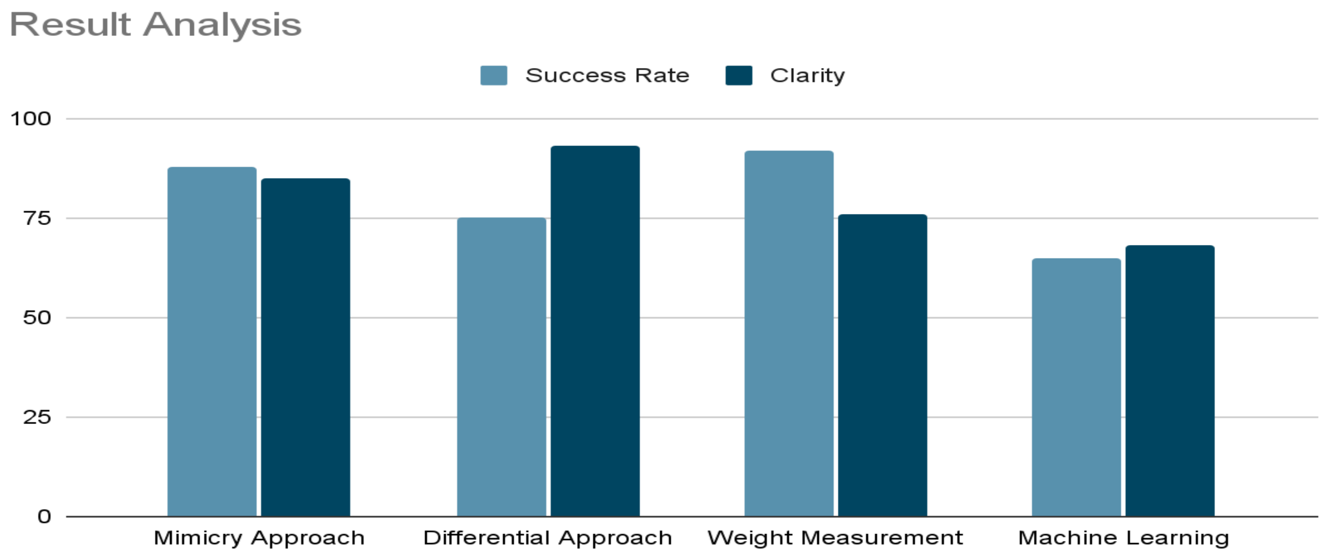
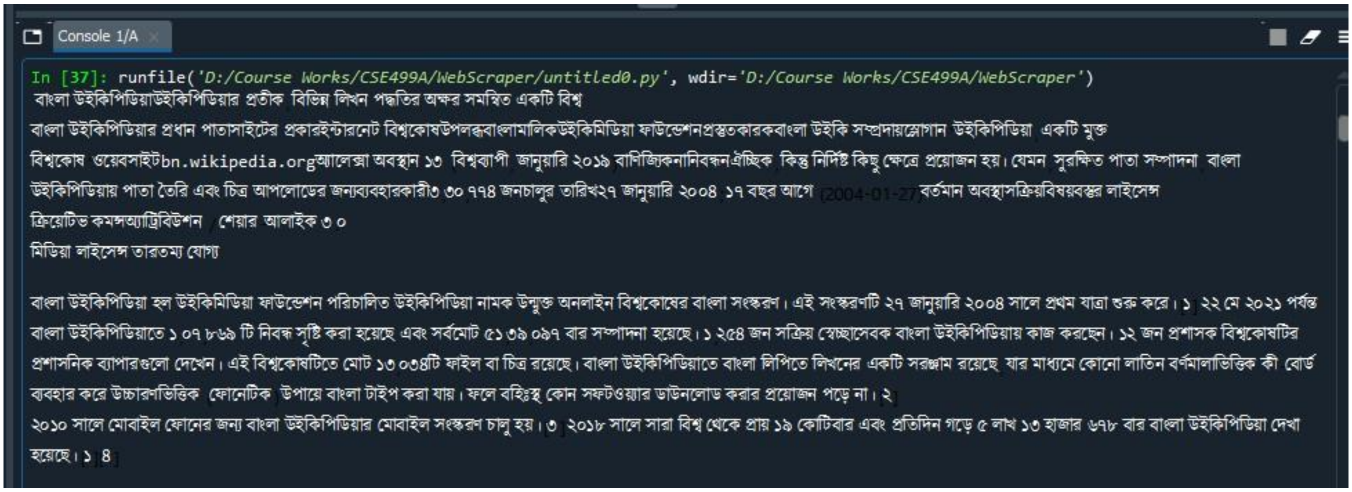


fig: Results analysis of 100 Bengali websites

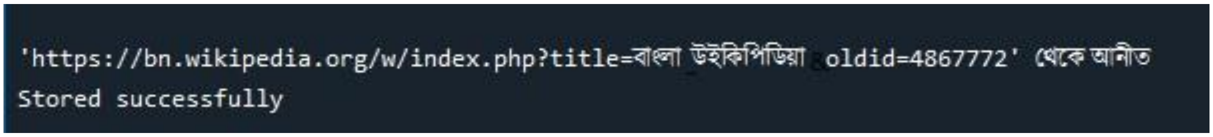
Results for storage: Part of project deals with storing the scraped and stored data in database automatically. It works as expected and is able to store the data in database successfully most of the times.

Scraped Content:



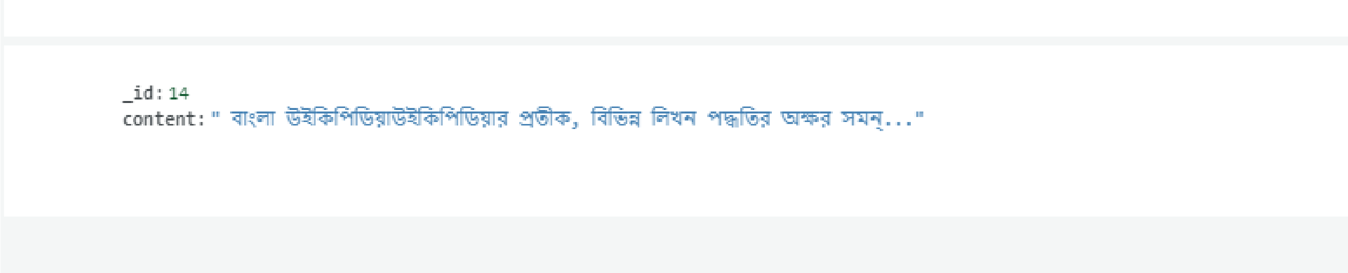
```
In [37]: runfile('D:/Course Works/CSE499A/WebScraper/untitled0.py', wdir='D:/Course Works/CSE499A/WebScraper')
 বাংলা উইকিপিডিয়াউইকিপিডিয়ার প্রতীক বিভিন্ন লিখন পদ্ধতির অক্ষর সমন্বিত একটি বিশ্ব
বাংলা উইকিপিডিয়ার প্রধান পাতাসাইটের প্রকারইন্টারনেট বিশ্বকোষউপলব্ধবাংলামালিকউইকিমিডিয়া ফাউন্ডেশনপ্রস্তুতকারকবাংলা উইকি সম্প্রদায়স্লোগান উইকিপিডিয়া একটি মুক্ত
বিশ্বকোষ ওয়েবসাইটbn.wikipedia.orgঅ্যালেক্সা অবস্থান ১৩ বিশ্বব্যাপী জানুয়ারি ২০১৯ বাণিজ্যিকনানিবন্ধনঐচ্ছিক কিন্তু নির্দিষ্ট কিছু ক্ষেত্রে প্রয়োজন হয়। যেমন সুরক্ষিত পাতা সম্পাদনা বাংলা
উইকিপিডিয়ায় পাতা তৈরি এবং চিত্র আপলোডের জন্যব্যবহারকারী৩ ৩০ ৭৭৪ জনচালুর তারিখ২৭ জানুয়ারি ২০০৪ ১৭ বছর আগে (2004-01-27)বর্তমান অবস্থাসক্রিয়বিষয়বস্তুর লাইসেন্স
ক্রিয়েটিভ কমন্সঅ্যাট্রিবিউশন /শেয়ার আলাইক ৩ ০
মিডিয়া লাইসেন্স তারতম্য যোগ্য

বাংলা উইকিপিডিয়া হল উইকিমিডিয়া ফাউন্ডেশন পরিচালিত উইকিপিডিয়া নামক উন্মুক্ত অনলাইন বিশ্বকোষের বাংলা সংস্করণ। এই সংস্করণটি ২৭ জানুয়ারি ২০০৪ সালে প্রথম যাত্রা শুরু করে।[১] ২২ মে ২০২১ পর্যন্ত
বাংলা উইকিপিডিয়াতে ১ ০৭ ৮৬৯ টি নিবন্ধ সৃষ্টি করা হয়েছে এবং সর্বমোট ৫১ ৩৯ ০৯৭ বার সম্পাদনা হয়েছে। ১ ২৫৪ জন সক্রিয় স্বেচ্ছাসেবক বাংলা উইকিপিডিয়ায় কাজ করছেন। ১২ জন প্রশাসক বিশ্বকোষটির
প্রশাসনিক ব্যাপারগুলো দেখেন। এই বিশ্বকোষটিতে মোট ১৩ ০৩৪টি ফাইল বা চিত্র রয়েছে। বাংলা উইকিপিডিয়াতে বাংলা লিপিতে লিখনের একটি সরঞ্জাম রয়েছে যার মাধ্যমে কোনো লাতিন বর্ণমালাভিত্তিক কী বোর্ড
ব্যবহার করে উচ্চারণভিত্তিক (ফোনেটিক) উপায়ে বাংলা টাইপ করা যায়। ফলে বহিঃস্থ কোন সফটওয়্যার ডাউনলোড করার প্রয়োজন পড়ে না।[২]
২০১০ সালে মোবাইল ফোনের জন্য বাংলা উইকিপিডিয়ার মোবাইল সংস্করণ চালু হয়।[৩] ২০১৮ সালে সারা বিশ্ব থেকে প্রায় ১৯ কোটিবার এবং প্রতিদিন গড়ে ৫ লাখ ১৩ হাজার ৬৭৮ বার বাংলা উইকিপিডিয়া দেখা
হয়েছে।[১][৪]
```

Database storing:



```
'https://bn.wikipedia.org/w/index.php?title=বাংলা_উইকিপিডিয়া&oldid=4867772' থেকে আনীত
Stored successfully
```

The code snippet output shows if we have managed to store the data successfully in database.



```
_id: 14
content: " বাংলা উইকিপিডিয়াউইকিপিডিয়ার প্রতীক, বিভিন্ন লিখন পদ্ধতির অক্ষর সমন্..."
```

This screenshot shows the successfully saved data in mongodb cluster.

CHAPTER 5: CONCLUSION

5.1 DISCUSSION

A few notable works related to our work was discussed earlier. Web scraping, however, is a somewhat common project. Other tools allow the scratching of data from any Web site like Classified sites, forums & E-Commerce scraping, PDFs, Simple Web pages. Our project is different from other scraper tools. We scrape only Bangla text articles from Bengali websites and try to find the barriers to scrape Bangla traditional websites and to fix this problems. This project aims to categorize the results and to store the discarded data so that it can be used in future projects. It also offers us a guideline and conclusion as to which strategy to scrape websites in Bengal would prove most efficient.

5.2 SUMMARY

Web Scraping is a great technique of extracting unstructured data from the websites and transforming that data into structured data that can be stored and analyzed in a database. Web Scraping is also known as web data extraction, web data scraping, web harvesting or screen scraping. Web scraping is a form of data mining. The overall goal of the web scraping process is to extract information from a websites and transform it into an understandable structure like spreadsheets, database or a comma-separated values (CSV) file. This project helps us find which algorithm and approaches can be used to scrape Bengali websites most accurately and efficiently.

5.3 FUTURE WORK

We would like to continue working on this project and hopefully turn it into a paper dedicated to finding out how the different approaches of web scraping gives us different results for scraping Bengali website and how it differs from traditional English website scraping approaches and their success rates. We would also like to increase the total number of tested websites and increase the sample size for the machine learning approach as well as create a web application that allows anyone to discover their query data in a more convenient manner.

APPENDIX

HTTP: Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML.

URL: The URL stands for Uniform Resource Locator, and it specifies the unique address for each internet resource.

Excel: For data analysis and documentation, Microsoft Excel is a useful and sophisticated software.

Parser: A parser is a component of a compiler or interpreter that breaks data down into smaller chunks for easier translation into another language. A parser takes a sequence of tokens, interactive commands, or computer instructions and breaks them down into pieces that can be used by other programming components.

CSV : A CSV (comma-separated values) file is a text file with a specified format for storing data in a table-structured format.

REFERENCES

[1] D. Glez-Peña, A. Lourenço, H. López Fernández, M. Reboiro Jato, and F. Fdez Riverola. Web Scraping Technologies in an API World. Briefings in Bioinformatics, 15 (5): 788-797, 2014. http://doi.org/10.1093/bib/bbt026

[2] N. R. Haddaway. The Use of Web-scraping Software in Searching for Grey Literature. *Grey Journal (TGJ)*, 11 (3), 2015.

[3] M. I. Varlamov and D. Y. Turdakov, ‘‘A survey of methods for the extraction of information from Web resources,’’ Program. Comput. Softw., vol. 42, no. 5, pp. 279–291, Sep. 2016.

[6] R. Mitchell, Web Scraping with Python: Collecting Data from the Modern Web. Newton, MA, USA: O’Reilly Media, 2015.

[7] https://www.researchgate.net/publication/324907302_Legality_and_Ethics_of_Web_Scraping

[8] O. Castrillo-Fernández, ‘‘Web scraping: Applications and tools,’’ Eur. Public Sector Inf. Platform, Spain, Topic Rep. 2015/10, Dec. 2015. [Online]. Available: https://www.europeandataportal.eu/sites/default/files/2015_web_scraping_applications_and_tools.pdf

[9] Re-fashioned IDC digital data-flood blather (https://theregister.com/2017/04/05/seagate_sponsors_refashioned_idc_digital_universe_blather)

[10] Beautiful Soup Documentation – www. crummy.com

[11] Crawling the Web, Gautam Pant, Padmini Srinivasan and Filippo Menczer.

[12] Web Crawler: A Review , Md. Abu Kausar , V. S. Dhaka Dept, Sanjeev Kumar Singh

[13] Web Scraping, Wikipedia.

[14] Text Categorization by Fabrizio Sebastiani Dipartimento di Matematica Pura e Applicata Universita di Padova ` 35131 Padova, Italy

[15] Webscrapper.io (https://webscraper.io/)

[16] Bar-llan, J. (2011). Data collection methods on the web for informetric purpose – A review and

analysis.

[17] Markus Herrmann, and Laura Hoyden, “Applied WebScraping in Market Research”, International Conference on Advanced Research Methods and Analytics, July 2016.