Web Scraping (Extraction of Deep Web Page Content)

Santosh G N[1] Mr. S. Lokesh[2]
[2]Associate Professor
[1]Department of Information Technology
[1,2]The National Institute of Engineering, Mysore

Abstract— Extracting useful information from the web is the most significant issue of concern for the realization of semantic web. This may be achieved by several ways among which Web Usage Mining, Web Scrapping and Semantic Annotation plays an important role. Web mining enables to find out the relevant results from the web and is used to extract meaningful information from the discovery patterns kept back in the servers. Web usage mining is a type of web mining which mines the information of access routes/manners of users visiting the websites. Web scraping, another technique, is a process of extracting useful information from HTML pages. The content is presented in a human-readable layout and is not intended to be processed by automatic systems. Therefore, it is necessary to separate the content in a web forum discussion from the layout before doing any further information mining. In this paper, explore and discuss some information extraction techniques on web like web usage mining, web scrapping for a better or efficient information extraction on the web illustrated with examples

Key words: HTML, Web Scrapping

I. INTRODUCTION

Web scraping is the process of automatically collecting useful information from web.It is also referred as web data extraction and extracts useful Information from HTML pages in various ways. Twitter is a micro blogging platform, limiting each entry to 140 characters. It's been described as the SMS of the Internet.

Like Twitter, there are several online micro blogging services like Identi.ca, Api.net, Google Buzz, etc. The message entry is called a Tweet and is publicly visible by default. Members can subscribe to others' tweets – this is known as following and subscribers are known as followers.

Twitter gives a couple of APIs to interact with its data: the REST API is useful for getting specific data and the Streaming API, which is good for real-time access to posts.

LinkedIn is a social networking website for people in professional occupations and is mainly used for professional social networking. LinkedIn has more than 200 million members in over 200 countries and territories, featuring micro blogging for its members. To access their data, LinkedIn offers a REST API.

Facebook is the largest social network to date having more than 1 billion members worldwide. Facebook incorporates micro blogging with features like "Wall" and "Status Updates". Like Facebook there are other online services such as Google+, MySpace, Diaspora, etc. Facebook provides the Open Graph API to access a member's data, but only through a Facebook registered application.

As the Internet gets bigger, while no official figures are given we can extrapolate that Google indexes more than 40 billion Webpages but identified more than a trillion unique URLs, and more people gain access to the Internet , more than two billion individuals, and these platforms, the generated content and information grows and a need to analyze and categorize this new trove of information increases.

A. Objective:

The objective of the project is to In order to scrape available data, it is necessary to perform: Preprocessing, Pattern Discovery, and Pattern Analysis. Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the web server logs, different web sites can help understand the user behaviour and the web structure, thereby improving the design of this colossal collection of resources.

B. Existing System:

Forum post extraction has been explored in recent works. Find the post region by comparing consecutive DOM nodes. The granularity of extraction is very coarse. For example, the author's information is extracted as a whole.

Particular entities, like the author's name are not considered. Use a visual approach to find the post region and boundaries between posts. However, only the post content is extracted. Other entities like the author's name or postdate are not considered. Present a site-level approach by rebuilding the sitemap of a forum.

This process needs approximately 2000 pages per domain. Similar to MDR, posts are found by comparing and aligning consecutive DOM trees within a thread page. Assumptions, like "if a post record has an author link, then it is the author node" are formulated in first-order logic and enriched.

C. Proposed System:

Web Scraping is a technique of automatic web data extraction to extract data from the HTML of website by parsing the Webpages using specially coded programs for manipulation such as converting the Web page to another format like XML or by embedding browsers. It is close to web indexing which indexes web content using softwood adopted by many search engines but it focuses more on the transformation of unstructured Web content into structured data that can be stored and analyzed in a central database/spreadsheet.

It's uses include online price comparison, weather data monitoring, website change detection, Web research, Web content mash up and Web data integration. It can provide various levels like: human copy and-paste, Text gripping (based on the UNIX command or regular expression matching facilities of programming languages like Perl), HTTP programming (HTTP requests to the

remote Web server), Embedding Web browsers programs can retrieve the dynamic contents generated by client side scripts), HTML parsers (to parse HTML pages and to retrieve and transform Web content) and Web scraping software tools . Prolog, a language used in (A.I) artificial intelligence, has the capability to interact with web server or web client , keep the required necessary data and extract the required information with the help of PSP(Prolog Server Pages) and some inference rules.

PSP accepts the arguments from HTML and generates the response (web server needed) and so it interacts with Prolog to generate an output and pass it to the HTML. What it needs is a web server, prolog compiler and a web browser. IIS (Internet Information Server) may be used as a web server to process the scripting language Prolog Server Pages and to produce the HTML response. Text grepping uses the regular expression matching technique where one tries to match a particular expression in the available file. After getting the suitable match, as per our expression.

II. SYSTEM DESIGN PHASE

We partition the system into several components – Crawl Managers, Harvesters, DNS Resolvers and the Crawl Applications. Splitting into components gives the ability to scale the system horizontally for increased performance and resilience. This way we can spawn instances depending on the job load as needed.

The communication between these components is done over TCP/IP using a job server for distributing the events triggered by any component. Gearman has been chosen for this application as it provides a generic application framework to farm out work to other machines or processes. For queue persistence, Redis is used. Redis is a high performance key value store. Using this combination, we have consistence on restarting or other interrupting events.

For storing files and metadata we chose a document oriented database. Because of the nature of the data crawled we cannot use a regular RDMS for this purpose as we have a dynamic number of attributes for each type of web sites. The chosen database is MongoDB.
Some of the features of MongoDB are:

- Full index support – can index on any attribute
- Auto-sharding – horizontal scaling
- Fast atomic updates –increased performance when updating records.
- Map/Reduce – provides flexible aggregation and data processing
- GridFS – storing files of any size

Text grepping uses the regular expression matching technique where one tries to match a particular expression in the available file. After getting the suitable match, as per our expression, we pick the values before or after this regular expression. Scraping program is required to update frequently from time to time due to which maintenance.

A. Crawl Application:

It's composed of several core components: URL Finder and Data Normalize. This is a basic form of a Crawl Application.

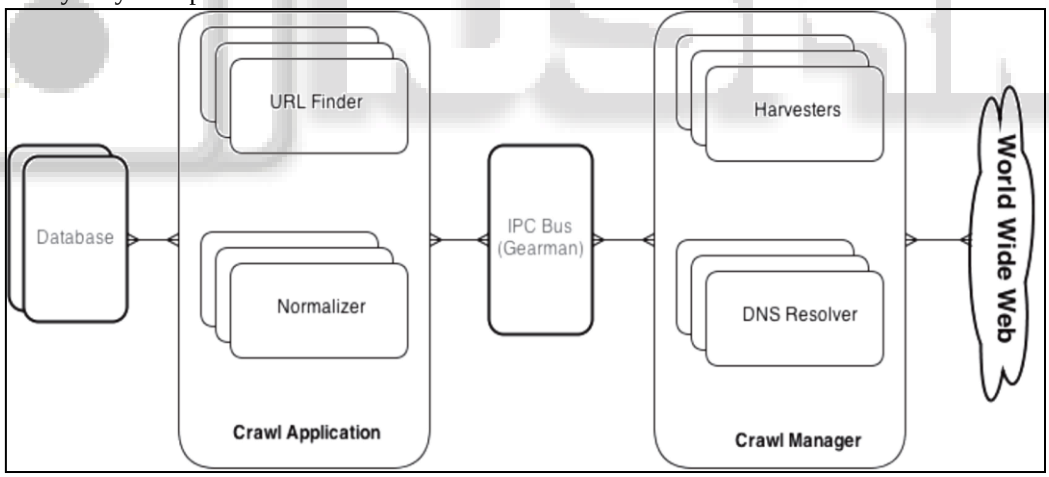

Fig. 1: Crawl Application and Manager Basic Components and Relations

The URL Finder parses the data stored in our database for new URLs and passes them to the Crawl Managers via the Job Queue. The Data Normaliser receives data from the Crawl Managers, for example a data feed or a web page, extracts the data (for example, extracting an article and comments from a blog post using heuristics). Many blogs put only abbreviated posts in their feeds to direct the user to their web site where they can monetize their efforts through ads for example. The normaliser extracts the meta-data (author, post date, title, image, etc.) from the feed and also tags the entry as incomplete if it finds that the post in the feed is abbreviated. Detecting this is done via a rule-based algorithm. It uses heuristics such as the length of the post in the feed, the presence of ellipsis at the end of the feed post, etc.

After the entire post page has been received from the crawler, the full blog post is extracted. A simple rule-based technique is used. The HTML page is loaded into DOM hierarchy and the textual node, which spans the entire post snippet from the feed, is identified. That node, most likely, contains the full blog post. All this processing is needed only for the blog post and not for its comments.

Going further, the Crawl Application tests the post or comment for spam. Social networks are not immune to messages containing spam. We take care of this by analyzing and filtering through a Bakes Classifier much like other anti-spam solutions. Unfortunately, spam comments

may have a negative impact on the blog post by making others comment on the spam rather than on the subject of the post.

On the full post, other meta-data can be extracted such as language used in the post using N-grams. By applying NLP algorithms we extract even more metadata such as objects, events, sentiment, etc.

Finally, the Crawl Application rebuilds the blogs network graph prior to committing the data to the database.

B. Crawl Manager:

The Crawl Manager is made up of Harvesters and DNS Resolvers. It manages the Harvester and DNS Resolver workers starting or stopping them on demand, queuing jobs for them, etc. Also, it is a bridge between Harvesters and the Crawl Application. When a request is received from the Crawl Application, it first checks the URL has been visited, if it's been visited and it's content is fresh it'll return a request to the Crawl Application – no need to send anything as we already have the data – else, if it's been visited and it's content is stale or if it's never been visited it's sent to the Harvesters to be retrieved.

Prior to actually sending the request to the Harvester workers, checks are being done using the sites robots directives and checking the resolved DNS address. The Crawl Manager does Domain and IP Based Throttling, pooling domains per IP address so that we wouldn't bother the web-servers at large. This is because either they may have a slow network connection and could be impacted in a negative way by our crawl, or may have some kind of intrusion detection system/denial of service detection – which our system could cause if left unattended – and cause some other trouble. Clustering the URLs per web host and giving the list to the same Harvester prevent this from happening. First resolving the domain names and clustering the results per IP address accomplishes it.

III. FUTURE ENHANCEMENT

We plan to combine the entity classification with a rule based approach. The extraction granularity will be refined from block segment level to token level. The results are going to be presented in further publications. Other issues are with on the fly generated web pages. For example, when loading a blog entry, the comments will be loaded in the browser from the same site or from another web service such as Disqus and will not be caught by our harvester. Adding exceptions for each issue that arises would create a maintenance nightmare. The solution would be implementing an artificial intelligence using Natural Language Processing and ML (machine learning) algorithms to continuously update the base model.

IV. CONCLUSION

The paper highlights with illustrations and experimental results the role of web usage mining, web scrapping in information extraction on web in a better and efficient way. Even we showed how to reconstruct a data schema for a large set of web forums almost automatically. We discussed how to find and extract entities within posts such as author, title, body text and publication date from all forums following the structure of a discussion board.

At the moment, our approach to data extraction in the Blog Crawler Application is based on standards and common observations. The downside is that we assume everyone abides by a common standard, which is not the case. Given that each service provider wants and needs ways to differentiate from others by creating their own micro-formats leads to fragmentation of standards.

REFERENCES

[1] Berners-LEE, T. , Hendler,J & Lassila, O, "The Semantic Web," Scientific American, May 2001.
[2] P. Lambrix, "Towards a Semantic Web for Bioinformatics using Ontology-based Annotation", in: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2005, pp. 3-7.
[3] Gu Chengjian, Huang Lucheng, "Web Mining in Technology Management, 2008 International Seminar on Business and Information Management978-0-7695-3560- 9/08, 2008 IEEEDOI 10.1109/ISBIM.2008.127
[4] Microsoft Academic Research, Cloud computing, http://libra.msra.cn/Keyword/6051/cloud-computing?query= cloud%20computing, 2012.
[5] Maurice de Kunder, (2013) "The Size of the WorldWideWeb",available from http://www.worldwidewebsize.com/
[6] Argaez E, (2012) "Usage and Population Statistics", available from http://www.internetworldstats.com/stats.htm
[7] Hurst M. and Maykov A., (2009) "Social Streams Blog Crawler", InProcedings of the 2009 IEEE Interntional Conference on Data Engineering
[8] Naghavi M. and Sharifi M., (2012) "A Proposed Architecture For Continuous Web Monitoring Through Online Crawling Of Blogs",International Journal of UbiComp (IJU), Vol. 3, No. 1
[9] Huang W., Zhang L., Zhang J., Zhu M., (2009) "Semantic Focused Crawling for Retrieving E-Commerce Information", Journal of Software, Vol. 4, No. 5

