Volume 5, Issue 4, April 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Intelligent Web Forum Crawling by Supervised Machine Learning Process

Naik Deepak Ranoji, Prof. Satish R. Todmal
Department of Computer Engineering
JSPM's Imperial College of Engineering and Research,Wagholi
Pune, Maharashtra, India

Abstract: Crawl relevant forum content from the web with minimum overhead is crawl by the supervised web-scale forum crawler. Forum threads have information content that is collected by forum crawlers. Web forum crawling problem to a URL-type have been reduced to recognition problem. It shows how to learn accurate and effective regular expression patterns of constant navigation paths by automatically created training sets using aggregated results from weak page type classifiers. Every forum have different layouts or styles and have different forum software packages, they always have similar constant navigation paths connected by specific URL types to direct users from entry pages to thread page. Robust page type classifiers can be get from as few as five annotated forums and applied to a large set of unseen forums. To have accurate specification we have used the machine learning process applied to large set of Forum.

Keywords—URL;Forum crawling; ITF Regex; URL Type;Page Type;Irobot;FoCUS;Crawler;Irobot;

I. INTRODUCTION

The World-Wide-Web is growing at a vast rate and it is becoming increasingly difficult to retrieve specific information on the web. The vast growth of the WWW poses unparallel scaling challenges for general purpose crawlers and search engines. In this paper, we present a supervised web-scale forum crawler with machine learning tendency known as Forum Crawler under Supervision (FoCUS),. The goal and objective of FoCUS is to crawl relevant forum content from the web with minimal overhead. It selectively seek out pages that are relevant to a predefined set of topics, rather than collecting and indexing all accessible web documents to be able to answer all possible ad-hoc queries by machine learning process. By checking into the database FoCUS continuously keeps on crawling the web and finds any new web pages that have been added to the web, pages that have been removed from the web. Growing and dynamic nature of the web has become a challenge to traverse all URLs in the web documents and to handle these URLs. We take here only one seed URL as input and search with a keyword, the searching result is based on keyword and it will fetch the web pages where it will find that keyword.

People can hold conversations in the form of posted messages on an Internet forum, or message board, is an online discussion site. It is approved by a moderator depending on the access level of a user or the forum setup; a posted message might need to be before it becomes visible. Forums have a specific set of pattern associated with them; e.g. a single conversation is called a "thread", or topic. The central part of the search engine i.e. Web Crawler which browses through the hyperlinks and stores the visited links for the future use.

The discussion forum is hierarchical or tree-like in structure: a forum can contain a number of sub-forums, each of which may have several topics. Within a forum's topic, each new discussion started is called a thread, which is replied by many people upon wish.

A forum consists of a tree like directory structure. The top end is "Categories" with different types of URL's. Again it is divided into sub-forums and these sub-forums can further have more sub-forums. It has graph structure. Three basic message board display formats: Non-Threaded/Semi-Threaded/Fully Threaded, are used with their own advantages and disadvantages.

A. Collection of Forum

In forums, index URLs, thread URLs, and page-flipping URLs have specific URL patterns. In this paper, by learning patterns of index URLs, thread URLs, and page-flipping URLs and adopting a simple URL string de-duplication technique which is performed by machine learning process. FoCUS can avoid repetition and duplication without duplicating detection Page Type.

B. Page Types

- Entry Page: The homepage of a forum, which contains a list of boards and is also the lowest common ancestor of all threads.
- Index Page: A page of a board in a forum, which usually contains a table-like structure; each row in it contains information of a board or a thread. Collect list-of- board page, list-of-board and thread page, and board page are all index pages.

- Thread Page: A page of a thread in a forum that contains a list of posts with user generated content belonging to the same discussion.
- Other Page: A page that is not an entry page, index page, or thread page.

C. Type of URL

There are four types of URL.

- Index URL: A URL that is on an entry page or index page and points to an index page. Its anchor text shows the title of its destination board.
- Thread URL: A URL that is on an index page and points to a thread page. Its anchor text is the title of its destination thread.
- Page-flipping URL: A URL that leads users to another page of the same board or the same thread. Correctly dealing with page-flipping URLs enables a crawler to download all threads in a large board or all posts in a long thread.
- Other URL: A URL that is not an index URL, thread URL, or page-flipping URL.
- EIT Path: An entry-index-thread path is a navigation path from an entry page through a sequence of index pages to thread pages.
- ITF Regex: An index-thread-page-flipping regex is a regular expression that can be used to recognize index, thread, or page-flipping URLs. FoCUS aims to learn ITF regex and applies directly in online crawling. The learned ITF regexes are site specific.

Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions.

D. ITF Regexes Learning

URL training sets goal is to construct automatically sets of highly precise index URL, thread URL, and page-flipping URL strings for ITF regexes learning.

FoCUS first learns a set of ITF regexes .Then it performs online crawling using a breadth-first strategy. FoCUS first pushes the entry URL into a URL queue; next it fetches a URL from the URL queue and downloads its page; and then it pushes the outgoing URLs that are matched with any learned regex into the URL queue. FoCUS repeats this step until the URL queue is empty or other conditions are satisfied. FoCUS does not need to group outgoing URLs, classify pages, detect page-flipping URLs, or learn regexes again for that forum.

Almost all search engines have highly optimized crawling system, although working and details of documentation of this system are usually with their owner. It is easy to create a crawler that would work slowly and download few pages per second for a short period of time. It’s a big challenge to build the perfect system design, I/O, network efficiency, robustness and manageability. Every search engine is divided into different modules among those modules crawler module is the module on which search engine depends the most because it helps to provide the best possible results.

II. LITERATURE SURVEY

A. Web Crawling

It is Automatic traversal of web to collect all the useful informative pages, effectively and efficiently gather information about link structure interconnecting the informative pages. Web application designed to manage user created content and store in Database. It is online discussion area where anyone can discuss their favorite topics.

- Working: Pre-samples few pages to discover the repetitive regions. Group pre-sampled pages into clusters based on their repetitive regions where each cluster can be considered a vertex in the sitemap.

B. Irobot

It is tool to crawl through Web Forums intelligently enough to understand structure of forums before selecting traversal path and search into the graph. It tends towards two issues are Important page and Important links.

III. SUPERVISE MACHINE LEARNING PROCESS

Machine learning can be considered a subfield of computer science and statistics. It has strong ties to artificial intelligence and optimization, which deliver methods, theory and application domains to the field.

Machine learning is employed in a range of computing tasks where designing and programming explicit, rule-based algorithms is infeasible. Example applications include spam filtering, optical character recognition (OCR), search engines and computer vision. Machine learning is sometimes conflated with data mining, although that focuses more on exploratory data analysis. Machine learning and pattern recognition "can be viewed as two facets of the same field.

Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time. A model is prepared through a training process where it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data. Example problems are classification and regression. Example algorithms are Logistic Regression and the Back Propagation Neural Network.

A. Steps in Supervise Learning

Supervised learning (machine learning) takes a known set of input data and known responses to the data, and seeks to build a predictor model that generates reasonable predictions for the response to new data.

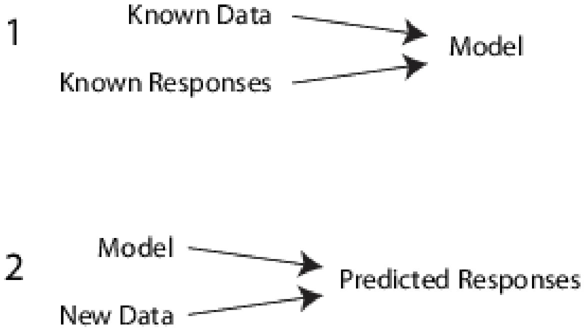

Fig 1. Supervise Learning

Suppose you want to predict if someone will have a heart attack within a year. You have a set of data on previous , including age, weight, height, blood pressure, etc. You know if the previous had heart attacks within a year of their data measurements. So the problem is combining all the existing data into a model that can predict whether a new person will have a heart attack within a year.

Supervised learning splits into two broad categories:

Classification for responses that can have just a few known values, such as 'true' or 'false'. Classification algorithms apply to nominal, not ordinal response values. Regression for responses that are a real number, such as miles per gallon for a particular car. You can have trouble deciding whether you have a classification problem or a regression problem. In that case, create a regression model first, because they are often more computationally efficient.

While there are many Statistics Toolbox algorithms for supervised learning, most use the same basic workflow for obtaining a predictor model. (Detailed instruction on the steps for ensemble learning is in Framework for Ensemble Learning.)

The steps for supervised learning are:

1. Prepare Data
2. Choose an Algorithm
3. Fit a Model
4. Choose a Validation Method
5. Examine Fit and Update Until Satisfied
6. Use Fitted Model for Predictions

SVM prediction speed and memory usage are good if there are few support vectors, but can be poor if there are many support vectors. When you use a kernel function, it can be difficult to interpret how SVM classifies data, though the default linear scheme is easy to interpret.

Naive Bayes speed and memory usage are good for simple distributions, but can be poor for kernel distributions and large data sets.

Nearest Neighbor usually has good predictions in low dimensions, but can have poor predictions in high dimensions. For linear search, Nearest Neighbor does not perform any fitting. For kd-trees, Nearest Neighbor does perform fitting. Nearest Neighbor can have either continuous or categorical predictors, but not both.

Discriminant Analysis is accurate when the modeling assumptions are satisfied (multivariate normal by class). Otherwise, the predictive accuracy varies.

B. Framework for Ensemble Learning

You have several methods for melding results from many weak learners into one high-quality ensemble predictor. These methods closely follow the same syntax, so you can try different methods with minor changes in your commands.

Create an ensemble with the fitensemble function. Its syntax is

ens = fitensemble(X,Y,model,numberens,learners)

1. X is the matrix of data. Each row contains one observation, and each column contains one predictor variable.
2. Y is the vector of responses, with the same number of observations as the rows in X.
3. model is a string naming the type of ensemble.
4. numberens is the number of weak learners in ens from each element of learners. So the number of elements in ens is numberens times the number of elements in learners.
5. learners is either a string naming a weak learner, a weak learner template, or a cell array of such templates.

Pictorially, here is the information you need to create an ensemble:

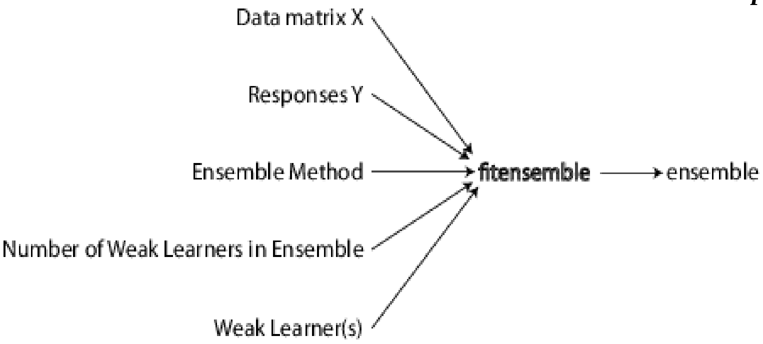

Fig 2. Ensemble Learning

IV. STUDIES AND FINDINGS

A. Read URL

We are concentrating on focus ontology which search for the relevant web pages based on the keyword we give. It forms a hierarchy of links. The web information on the particular web page for a particular keyword, which we give as, input is search. It goes for the link on that seed URL and after that switch to that link and find another link on that web page but it ,should match with the keyword, it will do like that until it reach the limit that we set. But it may be not possible that it will find the number of links that we set before. And it shows that the web page is not having any further link for that particular keyword. While fetching the links the user profiles also make sure that it should fetch only the unique links, i.e. it should not revisit the same link again and again. Finally, when we finished with the links, we will give one text file as input and run the three pattern matching algorithm.

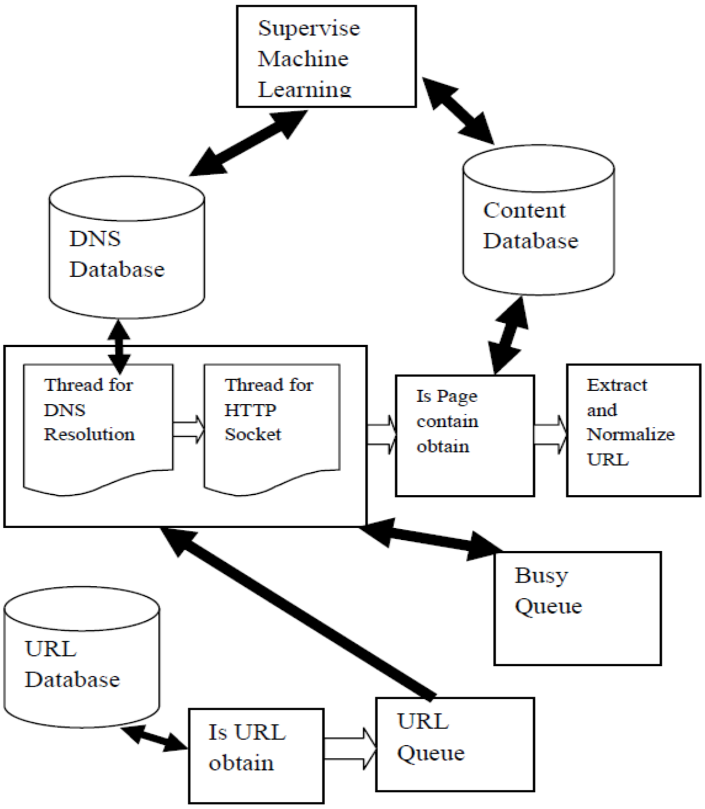

Fig 3. Architecture of Web Crawler by using Machine Learning

B. Pattern Recognition

Pattern recognition means only text. For syntax analysis Pattern matching is used. When we compare pattern matching with regular expressions then we will find that patterns are more powerful, but slow in matching. A pattern is nothing but character string. All keywords can be written in both the upper and lower cases. A pattern expression consists of atoms bound by unary and binary operators. Spaces and tabs can be used to separate keywords. Text mining is one of the important steps of knowledge discovery process. It extracts hidden information from not-structured or semi-structured data. It is fundamental because much of the web information is semi-structured due to the nested structure of HTML code, much of the web information is linked, and much of the web information is redundant. Web text mining helps in getting the whole knowledge mining process of mining, extraction and integration of useful data, information and knowledge from the web page content. Pattern recognition is applied on the web information in this manner, when we start the retrieval it will give me the links related to the keyword. It will then read the web pages that are extracted and retrieve from the links and while it will read the web page it will extract only the content. The contents are only the text that is available on the web page. It should not include images, tags, and buttons. It must be stored in some file and should not include any HTML tags.

C. Identification Process

The process will identify the required URL is whether right kind of link or wrong kind link. It will identify the URL, protocol link also for retrieve the relevant web page for user requesting the data. It's omits bad URLs while user requesting web pages. Bad URLs are identified by pattern of protocol occur on the relevant web pages on the server side store as database.

D. Downloading Process

After completion of all process the downloading will started by starting downloading requesting URL link of users need. After three checking process only it will download the relevant link according to users request. It will be working efficiently to users, the requested link will retrieve.

E. Index URL and Thread URL Training Sets

Recalling back; an index URL is a URL that is on an entry or index page; its destination page is another index page; its anchor text is the board title of its destination page. A thread URL is a URL that is on an index page; its destination page is a thread page; its anchor text is the thread title of its destination page. Difference between index URLs and thread URLs is the type of their destination pages. Therefore, it is necessary to decide the page type of a destination page. The index pages and thread pages each have their own typical layouts i.e. index page has many narrow records, relatively long anchor text, and short plain text; while a thread page has a few large records each post has a very long text block and relatively short anchor text. An index page or a thread page always has a timestamp field in each record, but the timestamp order in the two types of pages are reversed i.e. the timestamps are typically in descending order in an index page while they are in ascending order in a thread page.

The Index URL and Thread URL detection algorithm is given below.

1. Enter data
2. To collect all URL groups and longest link text length
3. Select URL group
4. IF the pages are not Index or Thread page then discarded.

F. Page-Flipping URL Training Set

Page-flipping URLs point to index pages or thread pages but are different from index URLs or thread URLs.

The "connectivity" metric to distinguish page-flipping URLs from other loop-back URLs but the metric only works well on the "grouped" page-flipping URLs, i.e., more than one page-flipping URL in one page.

In particular, the grouped page-flipping URLs have the following properties:

1. Their anchor text is either a sequence of digits such as 1, 2, 3, or special text such as "last."
2. They appear at the same location on the DOM tree of their source page and the DOM trees of their destination pages.
3. Their destination pages have similar layout with their source pages. We use tree similarity to determine whether the layouts of two pages are similar or not.
4. The single page-flipping URLs appearing in their source pages and their destination pages have the same anchor text but different URL strings.

Our page-flipping URL detection module works on above properties. The detail is shown in Fig. 3. Lines 1-11 first tries to detect the "group" page-flipping URLs; if it failed, lines 13-20 will enumerate all the outgoing URLs to detect the single page-flipping URLs; and line 23 set its URL type to page-flipping URL.

In our experiment over 160 forum sites (10 pages each of index and thread page), our method achieved 95 percent recall and 99 percent precision. We apply this method to both index pages and thread pages; they found page-flipping URLs are saved as training examples.

The Page Flipping URL detection algorithm is given below.

1. To detect the group page-flipping URLs if it fails.
2. It enumerates all the outgoing URLs to detect the single page-flipping URLs.
3. Set its URL type to page-flipping URL.

G. Evaluation of Entry URL Discovery

We assume that in forum crawling an entry URL is given. But finding forum entry URL has more value. For each forum in the test set, sampled page is fed to the module and it is manually checked if the output was indeed its entry page. In order to see whether low standard deviation indicates that it is not sensitive to sample pages. Two main failure cases: 1) forums are no longer in operation and 2) JavaScript generated URLs which we do not handle currently.

Entry URL Discovery algorithm:

1. URL check in all forum If keyword found ,the path from URL host.
2. Every page in a forum site contains a link to lead users back to its entry page.

3. URL is detected as an index URL
4. An entry page has most indexes URLs Since it leads users to all forum threads.

H. Evaluation of Online Crawling

In the previous sections we have shown that FoCUS is efficient in learning ITF regexes and is effective in detection of index URL, thread URL, page-flipping URL, and forum entry URL. This section compare FoCUS with other existing methods in terms of effectiveness and coverage .

V. CONCLUSION

We are working on focus crawler which search for the relevant web pages based on the keyword we give but it forms a hierarchy of links for the crawler on the particular web page for a particular keyword, which we give as, input. Crawler will search for the link on that seed URL and after that switch to that link and find another link on that web page but it should match with the keyword, it will do like that until it reach the limit that we set based upon the machine learning process. Partial-match, of Knutt-Morris-Pratt method identifies the bad URL in a website and number of character present in a web page. Focus identifies type of protocol used for the web page. And retrieve the web pages. We apply pattern recognition over text for correct navigation. Pattern symbolizes check text only i.e. what quantity text is available on web page.

REFERENCES

[1] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems, vol. 30, nos. 1-7, pp. 107-117, 1998.*

[2] C. Gao, L. Wang, C.-Y. Lin, and Y.-I. Song, "Finding Question- Answer Pairs from Online Forums," *Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 467-474, 2008.*

[3] Y. Guo, K. Li, K. Zhang, and G. Zhang, "Board Forum Crawling: A Web Crawling Method for Web Forum," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence, pp. 475-478, 2006.*

[4] M. Henzinger, "Finding Near-Duplicate Web Pages: A Large- Scale Evaluation of Algorithms," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 284-291, 2006.*

[5] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, "Deriving Marketing Intelligence from Online Discus- sion," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp.419-428, 2005.*

[6] H.S. Koppula, K.P. Leela, A. Agarwal, K.P. Chitrapura, S. Garg, and A. Sasturkar, "Learning URL Patterns for Webpage De- Duplication," *Proc. Third ACM Conf. Web Search and Data Mining, pp. 381-390, 2010.*

[7] K. Li, X.Q. Cheng, Y. Guo, and K. Zhang, "Crawling Dynamic Web Pages in WWW Forums," *Computer Eng., vol. 33, no. 6, pp. 80-82, 2007.*

[8] G.S. Manku, A. Jain, and A.D. Sarma, "Detecting Near-Duplicates for Web Crawling," *Proc. 16th Int'l Conf. World Wide Web, pp. 141- 150, 2007.*

[9] U. Schonfeld and N. Shivakumar, "Sitemaps: Above and Beyond the Crawl of Duty," *Proc. 18th Int'l Conf. World Wide Web, pp. 991- 1000, 2009.*

[10] X.Y. Song, J. Liu, Y.B. Cao, and C.-Y. Lin, "Automatic Extraction of Web Data Records Containing User-Generated Content," *Proc. 19th Int'l Conf Information and Knowledge Management, pp. 39-48, 2010.*

[11] "WeblogMatrix," http://www.weblogmatrix.org/, 2012.