Automated Data Mining from Web Servers Using Perl Script

Sandeep Neeli[†], Kannan Govindasamy[†], Bogdan M. Wilamowski[†], Aleksander Malinowski[‡]
[†]*Department of Electrical and Computer Engineering*
Auburn University, Auburn, AL, USA
neelisa, govinka, wilambm@auburn.edu
[‡]*ECE Department*
Bradley University, Peoria IL, USA
olekmali@bradley.edu

***Abstract*-Data mining from the Web is the process of extracting essential data from any web server. In this paper, we present a method called Ethernet Robot to extract information/data from a web server using perl scripting language and to process the data using regular expressions. The procedure involves fetching, filtering, processing and presentation of required data. The resultant HTML file consisting of the required data is displayed for the perusal of users. Future enhancements to our Ethernet Robot include optimization to improve performance and customization for use as a sophisticated client-specific search agent.**

***Keywords*-**Data Extraction, Data Mining, Perl, wget, Regular Expressions.

I. Introduction

The meteoritic rise of the World Wide Web as the knowledge powerhouse of the 21st century has led to a tremendous growth in information available to the masses. This, in turn, implies that the *useful* information is all the more time-consuming to narrow down or locate in the huge mass of available data. In other words, with increasing knowledge base, there is a pressing need to efficiently extract useful information in a shorter amount of time.

Retrieval of structured data from a pile of unstructured documents is called Data Extraction (DE). Web data extraction is a process of extracting information or data from the World Wide Web (WWW) and manipulating it according to the user constraints. A brief overview of web data extraction is discussed below and we present an example model of web data extraction based on these features in section II.

A. Phases of Automatic Web Data Extraction

The Web data extraction process can be divided into four distinct phases [20, 21, 26]:

1. Collecting Web Data – Includes past activities as recorded in Web server logs and/or via cookies or session tracking modules. In some cases, Web content, structure, and application data can be added as additional sources of data.
2. Preprocessing Web Data– Data is frequently pre-processed to put it into a format that is compatible with the analysis technique to be used in the next step. Preprocessing may include cleaning data of abnormalities, filtering out irrelevant information according to the goal of analysis, and completing the missing links (due to caching) in incomplete clickthrough paths. Most importantly, unique sessions need to be identified from the different requests, based on a heuristic, such as requests originating from an identical IP address within a given time period.
3. Analysing Web Data – Also known as Web Usage Mining [22, 23, 24], this step applies machine learning or Data Mining techniques to discover interesting usage patterns and statistical correlations between web pages and user groups. This step frequently results in automatic user profiling, and is typically applied offline, so that it does not add a burden on the web server.
4. Decision making/Final Recommendation Phase – The last phase in web data extraction makes use of the results of the previous analysis step to deliver recommendations to the user. The recommendation process typically involves generating dynamic Web content on the fly, such as adding hyperlinks to the last web page requested by the user. This can be accomplished using a variety of Web technology options such as CGI programming.

B. Categories of Data used in Web Data Extraction

The Web data mining process depends on one or more of the following data sources [25,26]:

1. Content Data – Text, images, etc, in HTML pages, as well as information in databases.
2. Structure Data –Hyperlinks connecting the pages to one another.
3. Usage Data – Records of the visits to each web page on a website, including time of visit, IP address, etc. This data is typically recorded in Web server logs, but it can also be collected using cookies or other session tracking tools.
4. User Profile – Information about the user including demographic attributes (age, population, etc), and preferences that are gathered either explicitly or implicitly.

C. Current Trend

Current tools that enable data extraction or data mining are both expensive to maintain and complex to design and use due to several potholes such as difference in data formats, varying attributes and typographical errors in input documents [1]. One such tool is an Extractor or Wrapper, which can perform the Data Extraction and processing tasks. Wrappers are special program routines that automatically extract data from Internet websites and convert the information into a structured format. Wrappers have three

978-1-4244-2083-4/08/$25.00 ©2008 IEEE

main functions.

- Download HTML pages from a website.
- Search, recognize and extract specified data.
- Save this data in a suitably structured format to enable further manipulation [2].

The data can then be further imported into other applications for additional processing.

Wrapper induction based on inductive machine learning is the leading technique available now a days.The user is asked to First label or mark the target items in a set of training pages or a list of data records in one page. The system then learns extraction rules from these training pages. Inductive learning poses a major problem - the initial set of labeled training pages may not be fully depictive of the templates of all other pages. Poor performance of learnt rules is experienced for pages that follow templates uncovered by the labeled pages. This problem can be solved by labeling more pages, because more pages cover more templates. Despite, manual labeling requires a large supply of labor and is time consuming with an unsatisfied coverage of all possible templates.

There are two main approaches to wrapper generation. The first and currently chief approach is wrapper induction. The second is automatic extraction. As discussed above, wrapper learning works as follows: The user first manually labels a set of training pages or data records in a list. A learning system then generates rules from the training pages. These rules can then be applied to extract target items from new pages. Sample wrapper induction systems include WIEN [9], Stalker [10, 11, 12], BWI [13], WL2 [14].
An analytical survey on wrapper learning [15] gives a family of PAC-learnable wrapper classes and their induction algorithms and complexities. WIEN [9] and Softmealy [16] are earlier wrapper learning systems, which were later improved by Stalker [11, 10, 17, 12]. Stalker learns rules for each item and uses more detailed depiction of rules. It treats the items separately instead of ordering them. Though this method is more flexile it makes learning harder for complex pages as the local information is not fully utilized. Recent developments on Stalker are the addition of different active learning facilities to the system which has reduced the number of pages which a user needs to label. Active learning allows the system to select the most useful pages which a user labels and hence reduces manual effort [18].
Other tools typically used are roadrunner [3], WebOQL [4], Automated Data Extraction by Pattern Discovery [5], etc.

Every day there is an exponential increase in the amount of information that seeps into the internet. Though this increases the possibility of finding a particular object, it also means a proportionate increase in search time. The tools for data extraction should therefore be developed with a view to reduce search time while keeping up with the internet advancements. In an attempt to serve this need, we present a new method of data extraction in this paper, called Ethernet Robot. Here, we make use of the Perl scripting language and the free non-interactive download utility- *wget.exe*. Features of *wget* are discussed in Section III. Notable features of the Perl language, which forms the core of our Ethernet Robot, is discussed below.

Perl is the most prominent web programming language available because of its text processing features and developments in usability, features, and/or execution speed. Handling HTML forms is made simple by the CGI.pm module, a part of Perl's standard distribution. It has the capability to handle encrypted Web data, including e-commerce transactions and can be embedded into web servers to speed up processing by as much as 2000%. The function "mod_perl" allows the Apache web server to embed a Perl interpreter [6]. Perl has a powerful regular expression engine built directly into its syntax. A regular expression or regex is a syntax that increases ease of operations that involve complex string comparisions, selections, replacements and hence, parsing. Regex are used by many text editors, utilities, and programming languages to search and manipulate text based on patterns. Regular expressions are widely used in our method to reduce the complexity of the code, to render the code obscure and powerful, and thus, unique. The combination of Perl, regular expressions and wget make Ethernet Robot an efficient solution for accelerated data downloading and extraction.

An overview of the proposed model for data extraction is given in Section II. Section III explains every stage of execution of the proposed model for a specific case with screenshots. We present our conclusions and recommendations for future work in Section IV.

II. Overview of Proposed Model

This section presents the proposed model of data extraction that can, in fact, draw only the necessary data from any web server on the internet. It can further be developed into a powerful search engine/portal.

Typically, a Data Extraction (DE) task is well-marked by its input and its extraction target. The inputs are usually unstructured documents like the semi-structured documents that are present on the Web, such as tables or itemized lists or a free text that is written in natural language [7].

Our proposed model of Data Extraction (DE), Ethernet Robot, can be used to download and extract any kind of information present on the internet according to the user requirements.

An Example

We consider an example of extracting titles and authors, pages, abstract URLs corresponding to the titles from the IEEE Transactions on Industrial Electronics located on IEEE Xplore. The main aim of this example is to allow the Associate Editors to search for reviewers, and Authors to search for paper references of the corresponding IEEE Transactions. The Transactions has papers listed according to the years of publication and each year has 6 issues. A screenshot of the transactions is shown in fig.1. In this figure, the boxes indicate the required data to be extracted and the inessential data or junk to be filtered out from each issue.. Lets now see how we automatically download and extract the data desired from these websites.

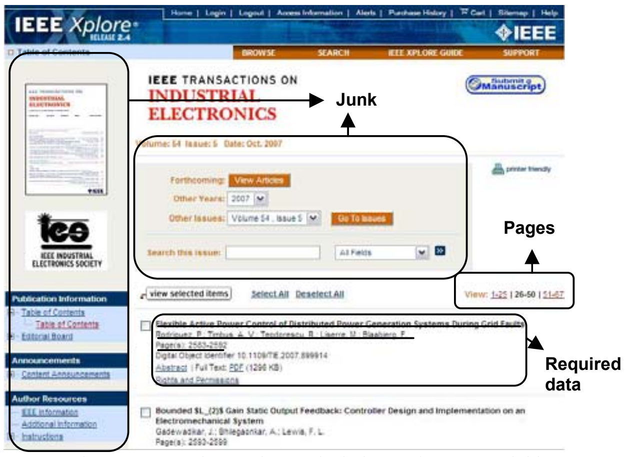


Fig. 1. IEEE Xplore webpage depicting various Data Fields.

Every Transactions on IEEE Xplore has a certain punumber, of which, Transactions on Industrial Electronics has a punumber = 41.
The generalized URL of an issue Z for the year/volume no.-Y is given according to IEEE as:

http://ieeexplore.ieee.org/servlet/opac?punumber=41&isvol=Y&isno=Z
If we want to download an extract the titles from volume no. 54 and issue 3 the URL is:

http://ieeexplore.ieee.org/servlet/opac?punumber=41svol=54sno=3
Again each issue may have sevaral pages 0,1,2,.. each page being addressed by :

http://ieeexplore.ieee.org/servlet/opac?punumber=Xisvol=Yisno=Z&page=P&ResultStart=Q
where page = P denotes the page number P, ResultStart=Q denotes the start of title number Q.
The URL -
http://ieeexplore.ieee.org/servlet/opac?punumber=41svol=54sno=3&page=1&ResultStart=25
is the link to the titles starting from number 26.
The page P=0 of any issue contains the links/URLs of the remaining pages as shown in Fig. 1. So the other pages can be fetched using the *wget* funtion and can be concatenated to the page P=0 to form a single page containing all the paper listings. Following script does the above process:

```
$p = $p0.$p1.$p2;
```

where $p0, $p1 and $p2 are the pages divided according to the paper listings and $p denotes the webpage containg all the paper listings of an issue. So the behavior of the tool is defined by the variables volume no.-Y, issue number - Z, page number – P. To download all the pages from years 2000 to 2006 say, the following conditions have to be included at the beginning of main perl code:

```
for($X=47, $x<=53, $x++)
{
for($Y=1,$Y<=6,$y++)
----
code
----
}
}
```

This example model of data extraction (Ethernet Robot) extracts all the titles and corresponding data from the IEEE Transactions on Industrial Electronics. In order to extract all data, the system needs to traverse all the paper list pages in the archive and then extract all the titles and data from each paper list page.
The code is devised to elicit the titles, authors, page numbers, abstract and abstract links from IEEE Xplore. Next, our Ethernet Robot goes to the webpage pointed by the URLs in each record and fetches the abstract. On completion of data acquisition, the raw data is printed in a new HTML file and published as a webpage.
The Ethernet Robot carries out four stages: Data collection, Data filtering, Data processing and Data presentation on web.
A schematic representation of the sequence of steps is given below:

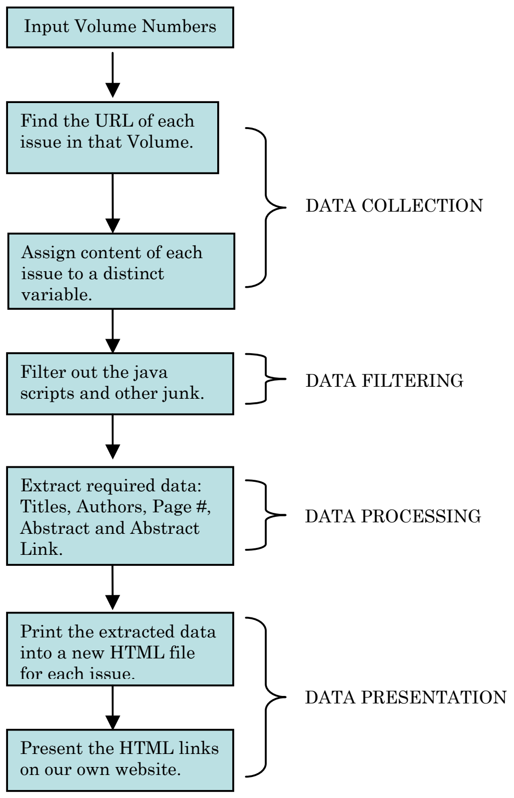

Input Volume Numbers
Find the URL of each issue in that Volume.
Assign content of each issue to a distinct variable.
DATA COLLECTION
Filter out the java scripts and other junk.
DATA FILTERING
Extract required data: Titles, Authors, Page #, Abstract and Abstract Link.
DATA PROCESSING
Print the extracted data into a new HTML file for each issue.
Present the HTML links on our own website.
DATA PRESENTATION

Fig. 2. Flowchart depicting the four stages of Ethernet Robot.

Of these, the data collection and filtering steps are relatively simple whereas the data processing and presentation steps require more involved procedures. These steps are explained in greater detail in the following sections.

III. Ethernet Robot

A. Data Collection

As mentioned in the previous section, the desired data to be fetched are specific volumes from the IEEE Transactions on Industrial Electronics. Thus, the starting point for this procedure is the Transactions webpage. Now, we invoke the function *get_page* with the parameter: *volume number*. *get_page* grabs the webpages corresponding to the *volume number* and returns one page per issue for each issue of the year/volume number.

The content of each issue is represented by a single variable -`$page`. Every year/volume has 6 issues, each of which is represented by a single element in an array of 6 variables. The following invariant holds true at any point during the operation of the code:

```
$p[$i]= $page for $i<=6
```

where	`$p`	– array of issue content
	`$i`	– issue number or iteration
	`$page`	– content of each issue

B. Data Filtering

Each webpage indicated by the variable `$p[$i]` consists of various irrelevant (to our purposes) javascripts, html tags, tables and other miscellaneous information appended to the data we wish to extract. Hence, the content of the page needs to be filtered. The following condition in the code performs the proposed filtering operation:

```
while ($issuepage =~
m/<table[^>]*>\s*<tr[^>]*>\s*<td[^>]*>\s*(.*
?)\s*<\/td>\s*<\/tr>\s*<\/table>/gis )

$entry = $1.
```

where the new variable '`$entry`' holds the required content between the `<table>` and `</table>`.

C. Data Processing

We now have the desired data in the variable `$issuepage`. All we need is to extract them in an orderly fashion in accordance with the IEEE format of representation. Following condition using the regular expressions divides the variables `$1` through `$6` into titles, authors, abstract links and pdf file links:

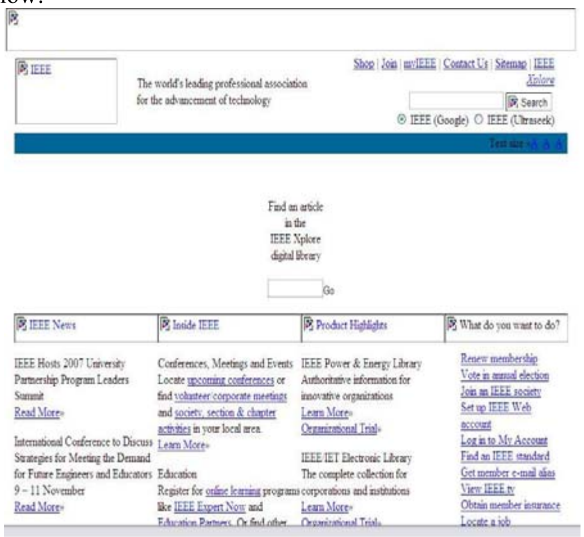
```
if($entry=~
m/<strong>(.*?)<\/strong>\s*<br>\s*((.*?)\s*
<br>)?\s*Page\(s\):&nbsp;.*<a\s+href="(.*?)"
>.*<a\s+href="(.*?)">.*<a\s+href="(.*?)">/is
)
```

`$title= $1`	- titles
`$authors= $3`	- authors
`$labs = $4`	- abstract links
`$lpdf = $5`	- pdf file links

We use wget.exe to obtain the abstracts from the abstract links. GNU Wget is a free utility for non-interactive download of files from the Web. It supports http, https, and ftp protocols, as well as retrieval through http proxies. Wget is non-interactive, which means that it can work in the background, while the user is not logged on. This allows the user to start retrieval and disconnect from the system, letting *wget* finish the work. In contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data [8]. The operation of wget.exe can be explained briefly as below:
The implementation of *wget* in perl is shown below to fetch the webpage addressing www.ieee.org :

```
use strict;
my $addr =
"http://www.ieee.org/portal/site";
system("wget.exe", "-q", "-O",
"example.htm", $addr);
```

The output of the above implementation is as displayed below:



IEEE

The world's leading professional association for the advancement of technology

Shop | Join | myIEEE | Contact Us | Sitemap | IEEE Xplore

Search

IEEE (Google) IEEE (Ultraseek)

Text size

Find an article in the IEEE Xplore digital library

Go

IEEE News

IEEE Hosts 2007 University Partnership Program Leaders Summit
Read More»

International Conference to Discuss Strategies for Meeting the Demand for Future Engineers and Educators 9 – 11 November
Read More»

Inside IEEE

Conferences, Meetings and Events
Locate upcoming conferences or find volunteer corporate meetings and society, section & chapter activities in your local area.
Learn More»

Education
Register for online learning programs like IEEE Expert Now and Education Partners. Or find other

Product Highlights

IEEE Power & Energy Library
Authoritative information for innovative organizations
Learn More»
Organizational Trial»

IEEE/IET Electronic Library
The complete collection for corporations and institutions
Learn More»
Organizational Trials

What do you want to do?

Renew membership
Vote in annual election
Join an IEEE society
Set up IEEE Web account
Log in to My Account
Find an IEEE standard
Get member e-mail alias
View IEEE.tv
Obtain member insurance
Locate a job

Fig. 3. Output of wget implementation in Perl. Example.htm.

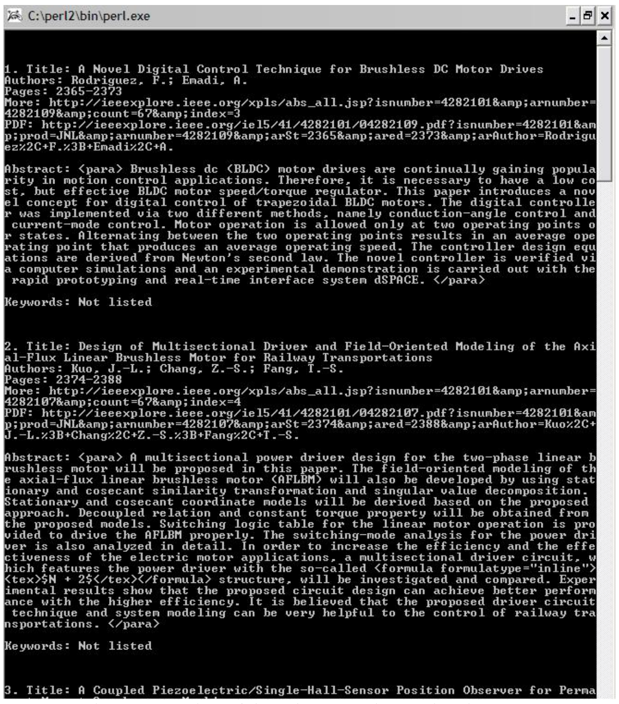
The essence of Ethernet Robot lies in the wget.exe function which downloads the desired data (Abstracts and pdf files) from the webpages. Wget.exe is responsible for the robotic behavior of our code as it does the automatic extraction of abstracts and pdf files.

The subfunction get_page makes use of wget.exe to extract all the data from a webpage.

```
sub get_page
{
my ($addr) = @_;
$addr =~ s/&amp;/&/g;
my $fname="54_1.htm";
system("wget.exe","-q","-O", $fname,"--
referrer  =http://tie.ieee-ies.org/tie/",
$addr);
my $page=get_file($fname);
unlink($fname);
return($page);
}
```

The wget parameter "`-O $fname`" specifies that the content in the webpage indicated by `$addr` will be printed to the file - `$fname` i.e., 54_1.htm.

Hence, at the end of execution of the code, we have all the required data corresponding to the issues in 6 separate files per year/volume. The execution of the perl code is shown in Fig. 4.

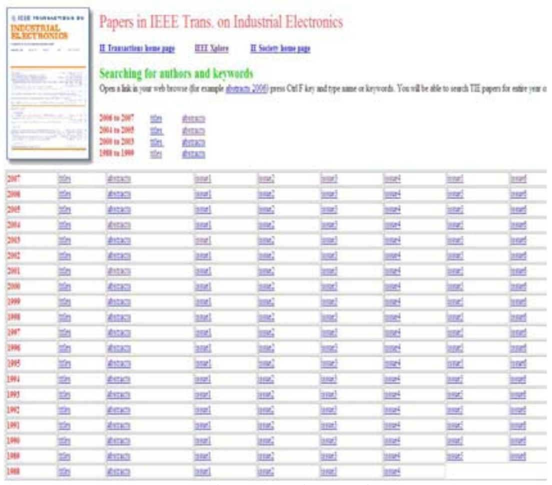


```
C:\perl2\bin\perl.exe

1. Title: A Novel Digital Control Technique for Brushless DC Motor Drives
Authors: Rodriguez, F.; Emadi, A.
Pages: 2365-2373
More: http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=4282101&amp;arnumber=
4282109&amp;count=67&amp;index=3
PDF: http://ieeexplore.ieee.org/iel5/41/4282101/04282109.pdf?isnumber=4282101&am
p;prod=JNL&amp;arnumber=4282109&amp;arSt=2365&amp;ared=2373&amp;arAuthor=Rodrigu
ez%2C+F.%3B+Emadi%2C+A.

Abstract: <para> Brushless dc (BLDC) motor drives are continually gaining popula
rity in motion control applications. Therefore, it is necessary to have a low co
st, but effective BLDC motor speed/torque regulator. This paper introduces a nov
el concept for digital control of trapezoidal BLDC motors. The digital controlle
r was implemented via two different methods, namely conduction-angle control and
 current-mode control. Motor operation is allowed only at two operating points o
r states. Alternating between the two operating points results in an average ope
rating point that produces an average operating speed. The controller design equ
ations are derived from Newton's second law. The novel controller is verified vi
a computer simulations and an experimental demonstration is carried out with the
 rapid prototyping and real-time interface system dSPACE. </para>

Keywords: Not listed


2. Title: Design of Multisectional Driver and Field-Oriented Modeling of the Axi
al-Flux Linear Brushless Motor for Railway Transportations
Authors: Kuo, J.-L.; Chang, Z.-S.; Fang, T.-S.
Pages: 2374-2388
More: http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=4282101&amp;arnumber=
4282107&amp;count=67&amp;index=4
PDF: http://ieeexplore.ieee.org/iel5/41/4282101/04282107.pdf?isnumber=4282101&am
p;prod=JNL&amp;arnumber=4282107&amp;arSt=2374&amp;ared=2388&amp;arAuthor=Kuo%2C+
J.-L.%3B+Chang%2C+Z.-S.%3B+Fang%2C+T.-S.

Abstract: <para> A multisectional power driver design for the two-phase linear b
rushless motor will be proposed in this paper. The field-oriented modeling of th
e axial-flux linear brushless motor (AFLBM) will also be developed by using stat
ionary and cosecant similarity transformation and singular value decomposition.
Stationary and cosecant coordinate models will be derived based on the proposed
approach. Decoupled relation and constant torque property will be obtained from
the proposed models. Switching logic table for the linear motor operation is pro
vided to drive the AFLBM properly. The switching-mode analysis for the power dri
ver is also analyzed in detail. In order to increase the efficiency and the effe
ctiveness of the electric motor applications, a multisectional driver circuit, w
hich features the power driver with the so-called <formula formulatype="inline">
<tex>$N + 2$</tex></formula> structure, will be investigated and compared. Exper
imental results show that the proposed circuit design can achieve better perform
ance with the higher efficiency. It is believed that the proposed driver circuit
 technique and system modeling can be very helpful to the control of railway tra
nsportations. </para>

Keywords: Not listed


3. Title: A Coupled Piezoelectric/Single-Hall-Sensor Position Observer for Perma
```

Fig. 4. Execution of the Ethernet Robot Perl code.

D. Data Presentation

The obtained files are then released into the World Wide Web by presenting them as links on a website. The data present in the issues can be further processed using perl to group all the issues of a year/volume in one whole file. Sample website which contains all the links for the desired data can be accessed from :
http://tie.ieee-ies.org/tie/abs/index.htm



Papers in IEEE Trans. on Industrial Electronics

IE Transactions home page IEEE Xplore IE Society home page

Searching for authors and keywords

Open a link in your web browse (for example abstracts 2006) press Ctrl F key and type name or keywords. You will be able to search TIE papers for entire year o

2006 to 2007	titles	abstracts
2004 to 2005	titles	abstracts
2000 to 2003	titles	abstracts
1988 to 1999	titles	abstracts

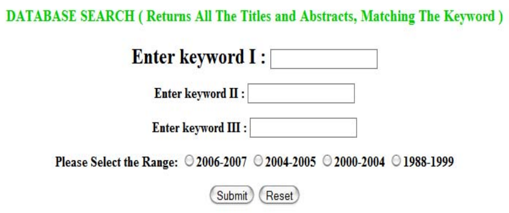
2007	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2006	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2005	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2004	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2003	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2002	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2001	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
2000	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1999	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1998	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1997	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1996	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1995	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1994	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1993	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1992	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1991	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1990	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1989	titles	abstracts	issue1	issue2	issue3	issue4	issue5	issue6
1988	titles	abstracts	issue1	issue2	issue3	issue4		

Fig. 5. Resultant webpage.

E. The Sorting Technique

After successfully extracting the data and storing them in our database, we further proceed to develop a search interface which can actually display all the Titles/papers for given Keywords like part of Title or Authors.

This tool is an addition to the Ethernet Robot, written in Perl/CGI which allows the user to search through the entire database we have extracted before and display the search results in a separate webpage. The search can be further refined or made selective by choosing the appropriate radio buttons for the corresponding years. Fig. 6 shows the search interface created for the above example.



DATABASE SEARCH (Returns All The Titles and Abstracts, Matching The Keyword)

Enter keyword I :

Enter keyword II :

Enter keyword III :

Please Select the Range: ○ 2006-2007 ○ 2004-2005 ○ 2000-2004 ○ 1988-1999

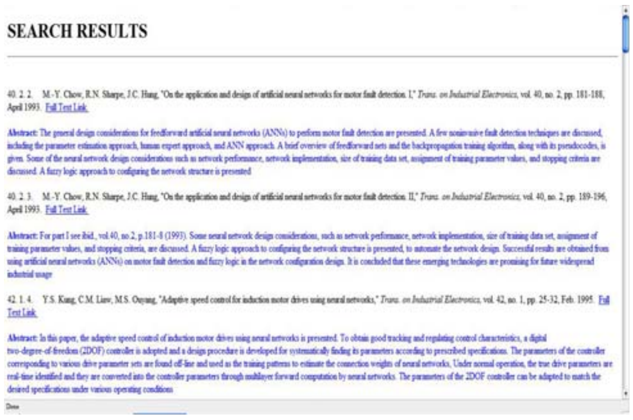
Fig. 6. Search Interface to download required papers.

This search tool can be included in the website hosted by our server by using Forms in the HTML page as follows:

```
<FORM METHOD="POST" ACTION="/cgi-
bin/examples/search.cgi"
.
.
</FORM>
```

In the above HTML script, *ACTION* tells the server to execute the *search.cgi* program on hitting the button *Submit.* The program *search.cgi* performs the data processing job again, fetching the files from the database on the server. *Param*, an inbuilt function in CGI script is used in this program to acquire user data from the HTML file.

For example, if the user mentions 'neural', 'networks' and 'motors' in the three query spaces- keyword I, keyword II and keyword III respectively, all the Titles and abstracts which have the keywords mentioned above will be displayed in a separate webpage. Fig. 7 shows the resultant webpage after performing the search operation.


SEARCH RESULTS

40. 2. 2. M.-Y. Chow, R.N. Sharpe, J.C. Hung, "On the application and design of artificial neural networks for motor fault detection. I," *Trans. on Industrial Electronics*, vol. 40, no. 2, pp. 181-188, April 1993. Full Text Link

Abstract: The general design considerations for feedforward artificial neural networks (ANNs) to perform motor fault detection are presented. A few noninvasive fault detection techniques are discussed, including the parameter estimation approach, human expert approach, and ANN approach. A brief overview of feedforward nets and the backpropagation training algorithm, along with its pseudocodes, is given. Some of the neural network design considerations such as network performance, network implementation, size of training data set, assignment of training parameter values, and stopping criteria are discussed. A fuzzy logic approach to configuring the network structure is presented

40. 2. 3. M.-Y. Chow, R.N. Sharpe, J.C. Hung, "On the application and design of artificial neural networks for motor fault detection. II," *Trans. on Industrial Electronics*, vol. 40, no. 2, pp. 189-196, April 1993. Full Text Link

Abstract: For part I see ibid., vol.40, no.2, p.181-8 (1993). Some neural network design considerations, such as network performance, network implementation, size of training data set, assignment of training parameter values, and stopping criteria, are discussed. A fuzzy logic approach to configuring the network structure is presented, to automate the network design. Successful results are obtained from using artificial neural networks (ANNs) on motor fault detection and fuzzy logic in the network configuration design. It is concluded that these emerging technologies are promising for future widespread industrial usage

42. 1. 4. Y.S. Kung, C.M. Liaw, M.S. Ouyang, "Adaptive speed control for induction motor drives using neural networks," *Trans. on Industrial Electronics*, vol. 42, no. 1, pp. 25-32, Feb. 1995. Full Text Link

Abstract: In this paper, the adaptive speed control of induction motor drives using neural networks is presented. To obtain good tracking and regulating control characteristics, a digital two-degree-of-freedom (2DOF) controller is adopted and a design procedure is developed for systematically finding its parameters according to prescribed specifications. The parameters of the controller corresponding to various drive parameter sets are found off-line and used as the training patterns to estimate the connection weights of neural networks. Under normal operation, the true drive parameters are real-time identified and they are converted into the controller parameters through multilayer forward computation by neural networks. The parameters of the 2DOF controller can be adapted to match the desired specifications under various operating conditions

Fig. 7. Webpage displaying the search results.

The user can right click the full text link to download and save the entire paper in pdf format. These links are provided by our server where all the extracted database is present.

For the further development of this technique, we can create an online tool using the same core concept and Google search, which when searched for an author or a title gives the details such as - total number of papers, total number of citations, average number of citations per paper, average number of citations per author, average number of papers per author, average number of citations per year, the age-weighted citation rate. This tool can also be used to find out the most cited paper and the most downloaded paper.

IV. Conclusions

This method of Data Extraction, Ethernet Robot, delivers optimum performance, making it a unique tool for web data extraction. The usage of perl scripting language, regular expressions and wget.exe make it accurate and advantageous. The Ethernet Robot can be customized according to the required data and the format of the data which a user desires. Future developments include creating a search engine which performs search throughout the database we have extracted before.

References

[1] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis and Khaled F. Shaalan, "A Survey of Web Information Extraction Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1411-1428, October 2006

[2] http://www.knowlesys.com/articles/web-data-extraction/wrapper_definition.htm

[3] Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," *Proc. the 26th Int'l Conf. Very Large Database Systems (VLDB)*, pp. 109-118, 2001.

[4] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs," *Proc. 14th IEEE Int'l Conf. Data Eng. (ICDE)*, pp. 24-33, 1998.

[5] C.-H. Chang, C.-N. Hsu, and S.-C. Lui, "Automatic Information Extraction from Semi-Structured Web Pages by Pattern Discovery," *Decision Support Systems J.*, vol. 35, no. 1, pp. 129-147, 2003.

[6] http://www.perl.org/about.html

[7] I-Chen Wu, Jui-Yuan Su, and Loon-Been Chen, "A Web Data Extraction Description Language and Its Implementation", *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05)*

[8] http://www.gnu.org/software/wget/manual/wget.html

[9] Kushmerick, N.: Wrapper induction for information extraction. PhD thesis (1997) Chairperson-Daniel S. Weld.

[10] Muslea, I., Minton, S., Knoblock, C.: A hierarchical approach to wrapper induction. In: AGENTS '99: *Proceedings of the third annual conference on Autonomous Agents. (1999)* 190-197

[11] Muslea, I., Minton, S., Knoblock, C.: Active learning with strong and weak views: A case study on wrapper induction. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*. (2003)

[12] Muslea, I., Minton, S., Knoblock, C.: Adaptive view validation: A first step towards automatic view detection. In: *Proceedings of ICML2002*. (2002) 443-450

[13] Freitag, D., Kushmerick, N.: Boosted wrapper induction. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. (2000) 577-583

[14] Cohen, W., Hurst, M., Jensen, L.: A flexible learning system for wrapping tables and lists in html documents. In: The Eleventh International World Wide Web Conference WWW-2002. (2002)\

[15] Kushmerick, N.: Wrapper induction: efficiency and expressiveness. Artif. Intell. (2000) 15-68

[16] Hsu, C.N., Dung, M.T.: Generating finite-state transducers for semi-structured data extraction from the web. Information Systems 23 (1998) 521-538

[17] Knoblock, C.A., Lerman, K., Minton, S., Muslea, I.: Accurately and reliably extracting data from the web: a machine learning approach. (2003) 275-287

[18] Yanhong Zhai and Bing Liu. "Structured Data Extraction from the Web based on Partial Tree Alignment" Accepted for publication in IEEE Transactions on Knowledge and Data Engineering, 2006

[19] http://ieeexplore.ieee.org/xpl/opac.jsp

[20] Schafer J.B., Konstan J., and Reidel J. (1999). Recommender Systems in E-Commerce, In *Proc. ACM Conf. E-commerce*, 158-166.

[21] Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on web usage mining, Communications of the. ACM, 43(8) 142–151.

[22] Spiliopoulou M. and Faulstich L. C. (1999). WUM: A Web utilization Miner, in *Proc. of EDBT workshop WebDB98*, Valencia, Spain.

[23] Nasraoui O., Krishnapuram R., and Joshi A. (1999). Mining Web Access Logs Using a Relational Clustering Algorithm Based on a Robust Estimator, 8th International World Wide Web Conference, Toronto, 40-41.

[24] Srivastava, J., Cooley, R., Deshpande, M., And Tan, P-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data, *SIGKDD Explorations*, 1(2), 12-23.

[25] Eirinaki M., Vazirgiannis M. (2003). Web mining for web personalization. *ACM Transactions On Internet Technology (TOIT)*, 3(1), 1-27.

[26] Malinowski and B.M. Wilamowski, "Compiling Computer Programs Through Internet", ITHET-2000 - International Conference on Information Technology Based Higher Education and Training, Istanbul, Turkey, July 3-5, 2000, pp. 343-348.

[27] Malinowski A. and B. M. Wilamowski, "Internet Accessible Compilers in Software and Computer Engineering" *International Conference on Simulation and Multimedia in Engineering Education (ICSEE'99)*, San Francisco, CA, January 14-15, 1999, pp. 221-224.