Similarity Based Web Data Extraction and Integration System for Web Content Mining

Srikantaiah K.C.[1], Suraj M.[2], Venugopal K.R.[1],
Iyengar S.S.[3], and L.M. Patnaik[4]

[1] Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering,
Bangalore University, Bangalore-560 001
[2] SJB Institute of Technology, Bangalore
[3] Florida International University, USA
[4] Indian Institute of Science, Bangalore
Srikantaiahkc@gmail.com

Abstract. The Internet is a major source of all information that we essentially need. The information on the web cannot be analyzed and queried as per the user requests. Here, we propose and develop a similarity based web data extraction and integration system (WDES and WDICS) to extract search result pages from the web and integrate its contents to enable the user to perform intended analysis. The system provides for local replication of search result pages, in a manner convenient for offline browsing. The system organizes itself into two possible phases that are involved in performing the above task. We develop and implement algorithms for extracting and integrating the content from the web. Experiment is performed on the contents of Bluetooth product listings and it gives us a better Precision and Recall than DEPTA [1].

Keywords: Offline Browsing, Web Data Extraction, Web Data Integration, World Wide Web, Web Wrapper.

1 Introduction

The World Wide Web has now become the largest knowledge base in the human history. The Web encourages decentralized authorizing in which users can create or modify documents locally, which makes information publishing more convenient and faster than ever. Because of these characteristics, the Internet has grown rapidly, which creates a new and huge media for information sharing and exchange.

There are situations in which the user needs those web pages on the Internet to be available offline for convenience. The reason being offline availability of data, limited download slots, storing data for future use, etc. This essentially leads to downloading raw data from the web pages on the Internet that is a major set of the inputs to a variety of software that are available today for the purpose of data mining.

In the recent years there has been lot of improvements on technology with products differing in the slightest of terms. Every product needs to be tested thoroughly, and

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 269–274, 2012.

internet plays a vital role in gathering of information for the effective analysis of the products.

Several algorithms have been proposed to extract data from search engine result pages which contains both structured data and unstructured data. ANDES [2] uses XML technology for data extraction and provides access to the deep web. Xunhua Liu et al., [3] have proposed an algorithm based on the position of DIV to extract main text from the body of Web pages. DEPTA [1] performs web data extraction automatically in two steps, in first step identifies the individual records in a page based on visual information and DOM tree matching. In second step aligns and extracts data items from the identified records based on partial alignment technique.

ONTOWRAPPER [4] is an ontological technique uses existing lexical database for English for the extraction of data records from deep web pages. Chia-Hui Chang et al., [5] have surveyed the major web data extraction approaches and compared them in three dimensions: the task domain, the automation degree, and the technique used. In these methods, the page containing required data is crawled[6] and then it is processed through online. This leads to a problem of offline unavailability of data, limited download slot etc., it can be overcome by using offline browsing mechanism [7].

In our approach, we replicate search result pages locally based on comparing page URLs with a predefined threshold. The replication is such that the pages are accessible locally in the same manner as on the web. In order to make the data available locally to the user for analysis we extract and integrate the data based on the prerequisites which are defined in the configuration file.

Contribution: In a given set of web pages, it is difficult to extract matching data. so, we have to develop a tool that is capable of extracting the exact data from the web pages. In this paper, we have developed WDES algorithm, which provides offline browsing of the pages. Here, we integrate the downloaded content onto a defined database and provide a platform for efficient mining of the data required.

2 Proposed Model and Algorithms

2.1 Problem Definition

Given a start page URL and a configuration file, the main objective is to extract pages which are hyperlinked from the start page and integrate the required data for analysis using data mining techniques. The user has sufficient space on the machine to store the data that is downloaded.

2.2 Mathematical Model

Web Data Extraction Using Similarity Function (WDES): A connection is been established to the given url *S* and the page is processed with the parameters obtained from the configuration file *C*. On completion of this, we obtain the web document that contains the links to all the desired contents that are obtained out of the search performed. The web document contains individual sets of links that are displayed on

each of the search results pages that are obtained. For example, if a search result obtained contains 150 records displayed as 10 records per page (in total 15 pages of information), we would have 15 sets of web documents each containing 10 hyperlinks pointing to the required data. This forms the set of web documents, *W*. i.e.,

$$W = \{w_i : 1 \leq i \leq n\}. \tag{1}$$

Each web document $w_i \in W$ is read through to collect the hyperlinks that are contained in it, that are to be fetched to obtain the data values. We, represent this hyperlink set as *H(W)*. Thus, we consider *H(W)* as a whole set containing all the sets of hyperlinks on each page $w_i \in W$. i.e.,

$$H(W) = \{H(w_i) : 1 \leq i \leq n\}. \tag{2}$$

Then, considering each hyperlink $h_j \in H(w_i)$, we find the similarity between h_j and *S*, using equation (3)

$$SIM(h_j, S) = \frac{\sum_{i=1}^{min(nf(h_j), nf(S))} fsim(f_i h_j, f_i S)}{(nf(h_j), nf(S))/2}. \tag{3}$$

where *nf(X)* is the number of fields in *X* and $fsim(f_i h_j, f_i S)$ is defined as

$$fsim = \begin{cases} 1 \; if \; f_i h_j = f_i S \\ 0 \; if \; f_i h_j = f_i S \end{cases} \tag{4}$$

The similarity $SIM(h_j, S)$ is the value that lies between 0 and 1, this value is used to compare with the defined threshold T_o (0.25), we download the page corresponding to h_j to local repository if $SIM(h_j, S) \geq T_o$. The detailed algorithm of WDES is given in Table 1.

The algorithm WDES navigates the search result page from the given URL *S* and configuration file *C* and generates a set of web documents *W*. Next, call the function *Hypcollection* to collect hyperlinks of all pages in w_i, indexed by $H(w_i)$, page corresponding to $H(w_i)$ is stored in the local repository. The function *webextract* is recursively called for each $H(w_i)$. Then, for each $h_i \in H(w_i)$, similarity between h_i and *S* is calculated using Eq. 3, if $SIM(h_i, S)$ is greater than the threshold T_o, then page corresponding to h_i is stored and collect all the hyperlinks in h_i to *X*. Continue this process for *X*, until it reaches maximum depth *l*.

Web Data Integration using Cosine Similarity(WDICS): The aim of this algorithm is to extract data from the downloaded web pages (those web pages that are available in the local repository i.e., output of WDES algorithm) into the database based on attributes and keywords from the configuration file C_i. We collect all result pages *W* from local repository indexed by *S*, then *H(W)* is obtained by collecting all hyperlinks from *W*, considering each hyperlink $h_j \in H(w_i)$ such that $k \in keywords$ in C_i. On existence of *k* in h_j, we populate the new record set *N[m, n]* by passing page h_j and

obtaining values defined with respect to the *attributes[n]* in C_i. We then populate the old record set *O[m, n]* by obtaining all values with respect to *attributes[n]* in database. For each record *i*, $1 \leq i \leq m$ we find the similarity between *N[i]* and *O[i]* using cosine similarity

$$SimRecord(N_i, O_i) = \frac{\sum_{j=1}^{n} N_{ij} O_{ij}}{\sqrt{\sum_{j=1}^{n} N_{ij}^2 \sum_{j=1}^{n} O_{ij}^2}}. \quad (5)$$

If similarity between records is equal to zero, then we compare each *attribute[j]* $1 \leq j \leq n$ in the records and form *IntegratedData* with use of Union operation and store in the database. The detailed algorithm of WDICS is shown in Table 2.

$$\text{IntegratedData} = \text{Union}(N_{ij}, O_{ij}). \quad (6)$$

Table 1. Algorithm: Web Data Extraction using Similarity Function (WDES)

```
Input
S : Starting Page URL. C: Parameter Configuration File.
l : Level of Data Extraction. T_o: Threshold.
Output: Set of Webpages in Local Repository.
begin
  W=Navigate to Web document on Given S and automate page with C
  H(W)=Call: Hypcollection(W)
  for each H(w_i) ∈ H(w)
      Save page H(w_i) on local Machine page P
      Call: Webextract(H(w_i), 0, pageppath)
  end for
end
Function Hypcollection(W)
begin
  for each w_i ∈ W do
      H(w_i)=Collect all hyperlinks in w_i
  end for
  return H(W)
end
Function Webextract(Z, cl, lp)
Input
Z : set of URLs. cl : Current level. lp : local path to Z.
Output: Set of Webpages in Local Repository.
begin
  for each h_i ∈ Z do
    if SIM(h_i, S) • T_o then
       Save h_i to Fh_i
       X=collect URLs from h_i and change its path in lp
       if( cl < l)
          Call: Webextract(X, cl + 1, pageppath of X)
      end if
    end if
  end for
end
```

3 Experimental Results

The experiment was conducted on the Bluetooth SIG Website [8], which contains listings of Bluetooth products and its specifications and is a domain specific search engine. We have collected data from www.bluetooth.org, which gives the listings of qualified products of Bluetooth devices. Here, we have extracted pages on the date range Oct-2005 to Jun-2011 consisting of total 92 pages, with each page containing 200 records of information. We were able to extract data from each of these pages. Based on the data extracted on the given attribute mentioned in the configuration file, we have a cumulative set of data for comparison.

Table 2. Algorithm: Web Data Integration using Cosine Similarity (WDICS)

```
Input
S : Starting Page URL stored in local repository (output of
    WDES).
C_i : Configuration File (Attributes and Keywords).
Output: Integrated Data in Local Repository.
begin
  H(w)=Call: Hypcollection(S)
  for each H(w_i) ∈ H(w) do
     Call: Integrate(H(w_i))
  end for
end
Function Integrate(X)
Input: X : set of URLs.
Output: Integration of Values of Attributes Local Repository.
begin
  for each hi ∈ Z do
    if( h_i contain keyword) then
      new[m][n]=parse page to obtain values of defined
                attributes[n] in C_i
      old[m][n]=obtain all values of attributes[n] from
                repository
      for each record i do
        if(SimRecord(new[i], old[i])==1) Skip
        end if
        else
        for each attribute j do
           if ( new[i][j] Not Equal to old[i][j] )
               IntegratedData=union(new[i][j],old[i][j])
           end if
        end for
        store IntegartedData in local repository
      end for
      X=collect all links for h_i
      if (X not equal to NULL) Call: Integrate(X)
      end if
    end if
  end for
end
```

The Precision and Recall are calculated based on the total available records in the Bluetooth website, the records found by the search engine and the records extracted by our model. Recall and Precision of DETPA are 98.67.1% and 95.05% respectively, and that of WDICS is 99.77% and 99.60% respectively as shown in Table 3. WDICS is more efficient than DEPTA because when an object is dissimilar to its neighboring objects DEPTA failed to identify all records correctly.

Table 3. Performance Evaluation between WDICS and DEPTA

Attributes	Total Records(TR)	DEPTA		WDICS	
		Extracted Records(ER)	Correct Records(CR)	Extracted Records(ER)	Correct Records(CR)
Name	18234	18204	17325	18234	18234
Model	18234	17860	17010	18060	18060
Company	18234	18095	17208	18234	18198
Spec Version	5508	5410	5016	5508	5426
Product Type	18234	17834	17015	18234	18045
Total	78444	77403	73574	78270	77963
Recall=(ER/TR)*100		98.67%		99.77%	
Precision=(CR/ER)*100		95.05%		99.60%	

4 Conclusions

Extraction of exact information from the web is an important issue in web mining. We propose a Similarity based Web data Extraction and Integration System (WDES and WDICS). The proposed approach includes extraction and integration of web data. This provides faster data processing and effective offline browsing functionality that helps in saving time and resource. Integrating onto the database helps in extracting the exact content from the downloaded pages.

References

1. Yanhong, Z., Bing, L.: Structured Data extraction from the Web Based on Partial Tree Alignment. Journal of IEEE TKDE 18(12), 1614–1627 (2006)
2. Jussi, M.: Effective Web Data Extraction with Standards XML Technologies. In: Proceedings of the 10th International Conference on World Wide Web, pp. 689–696 (2001)
3. Xunhua, L., Hui, L., Dan, W., Jiaqing, H., Wei, W., Li, Y., Ye, W., Hengjun, X.: On Web Page Extraction based on Position of DIV. In: IEEE 4th ICCAE, pp. 144–147 (2010)
4. Hong, J. L.: Deep Web Data Extraction. In: IEEE International Conference on Systems Man and Cybernetics (SMC), pp. 3420–3427 (2010)
5. Chia-Hui, C., Moheb Ramzy, G.: A Survey of Web Information Extraction Systems. Journal of IEEE TKDE 18(10), 1411–1428 (2006)
6. Tiezheng, N., Zhenhua, W., Yue, K., Rui, Z.: Crawling Result Pages for Data Extraction based on URL Classification. In: IEEE 7th Web Information Systems and Application Conference, pp. 79–84 (2010)
7. Ganesh, A., Sean, B., Kentaro, T.: OWEB: A Framework for Offline Web Browsing. In: Fourth Latin America Web Congress. IEEE Computer Society (2006)
8. Bluetooth SIG Website, `https://www.bluetooth.org/tpg/listings.cfm`