HTML based Ontology Extraction Semantic web: An Automated Approach

Anis Vosoogh\(^1\)* and Fatemeh Rezaeimehr\(^2\)

\(^1\)Young Researchers and Elite Club, Sirjan Branch, Islamic Azad University, Sirjan, Iran

\(^2\)Department of computer engineering, University of Kurdistan, Sanandaj, Iran

*Corresponding Author's E-mail: Anis.Vosoogh@iausirjan.ac.ir

Abstract

In this paper we focused on ontology extraction form web domains that as far as possible to doing automatically and accurately. For this we proposed a new method transfer HTML documents to ontology layer. This web domain-based ontology where has named WDO used of new parsing HTML documents and domain features for improving attained ontology. We have running our method for extract ontology from university domain as a distinct domain. The experimental results where are conflicted by Persian language that has been shown our work is applicable. Finally, we have evaluated WDO with the other methods for ontology extraction.

Keywords: Semantic web, ontology extraction, content management system, HTML parsing.

1. Introduction

Web is considered as one of the important sources of information. With increasing development of web, the problem of effective semantic processing of such a type of information is introduced. When there are several thousand petabyte of information on the web, retrieving the required information and different targeted processing on it are difficult. The problem originates from the point that structural layer of web database is separated from knowledge layers, containing the content of such database. This makes its processing difficult for machine. However, the information about every topic is continuously expanding. Therefore, achieving the most accurate of them in a short time from the view of a user is considered as the current challenge of researchers in the web. This problem makes a specific condition for search engines [1]. Semantic web has been introduced to make information significant for processing and extracting information. This area of research in the web has been introduced after creating technologies such as xml [2], and makes the data on the web understandable for web machines by tagging them and creates every type of semantic relationship which is not conceivable through structural relationship [3]. Semantic web has focuses on the ontology extraction [4]. Ontology extraction is considered as attractive studies in the field of semantic web. Nevertheless, there is no considerable attention on the Persian web and extract ontology from it. Current study aims to introduce a method to extract ontology from web domains when Persian langue is an alternative for contents.

II. RELATED WORKS

As a great source of expanding information, web needs some useful interfaces to response the data searching requests, data exchange, and even using data for different purposes. This needs employment of semantic web technology to upgrade the current web. Realization of semantic web occurs in the platform of frames such as mining and dissemination of available domains ontology. Presentation of
current knowledge of web at the level of meaningful ontology, determine the web information for machine and enable it in different applications. For example, specific search for a topic among published data in relational bases available in web is considered as a public demand in different frames such as search services, automatic indexing, automatic link exchange, etc. Nevertheless, quality and accuracy of available approaches is not satisfied [5]. Importance of addressing the issue and its difficulty become clearer when client-side differences such as type of language, current time-dependent concepts and much different applications are considered on the information available in relational web databases. Therefore, it is necessary to consider effective applied approaches for creation of semantic web layer and facilitation of web services especially based on users’ needs. Ontology extraction is considered as the most important stage of this process for information available on relational databases (RDB). However, as it is presented in part 3-9-2, ontology extraction from HTML documents has been partially studied. Nevertheless, it is necessary to offer some suggestions for possible approaches of ontology extraction from web beside most theoretical approaches [8, 7], and or special purpose [9, 10].

Ontology extraction methods for relational databases often are effective when they have access to a relational database schema [8,6]. In a general view, this will make many limitations in semantic web. For example, if extract ontology domain of Iran universities and form its semantic web layer, it must provide access to relational database scheme for more than 1000 different universities, while this is impossible. Although, even if somehow managing this problem resolve, ontology extraction approaches based on relational scheme haven’t high performance to present answer at the level of a semantic web service [9]. Perhaps the most important problem to represent a practical approach to create ontology domain is that most of databases information are not available. However, this is not the only problem. Irrelevant information, permanent change in pages, and high processing overhead are problems which make the semantic web ontology extraction hard on available RDBs. In this chapter we will illustrate a new approach of domain ontology extraction when there is no access to relational database scheme.

The method can support ontology domain extraction semi-automatically. Efficiency of the approach is limited to type of managing HTML assignments related to a domain. In the suggested approach, when dynamic construction of HTML pages follows a template (same or different) with content of data available in relational web database, extracted ontology will provide a high assessment features. Today, modeling of HTML documents structures is very popular. Due to high rate of webpages construction, currently all the process of creation, publication and change of HTML documents is carried by content management software (CMS). Therefore, many different domains of Web pages have a common pattern. Hence, based on the features CMS provide for management of webpages, mapping method of CMS to ontology is introduced as making the ontology by scheme-like RDB and HTML tree (PSTH)1 .

Concentration on the part of web which its documents are modeling doesn’t mean the capability of suggested approach in ontology extraction; even most of today’s web is so [5,4]. Today, a large portion of databases available in web are managed by portals, which CMS is part of it. We will study the structure of management of HTML pages in general condition and when are managed by CMSs. We will indicate how to find a way to extract ontology automatically from databases based on CMSs. Then, we will suggest a new architecture to create a semi-automatic ontology from webpages. Investigation of the benefits of suggested approach in contrast to other approaches will state the possible platforms for evaluation of this approach.

III. HTML BASE ONTOLOGY EXTRACTION

Lack of direct access to the database schema is the main problem of ontology extraction from relational database in web. Therefore, natural construction and publishing ontology of these databases which includes a huge mass of information must conduct and publish through webmasters that can access to databases or followed through crawling HTML pages. If we want to present an automatic approach to extract ontology from RDB related to specified domain publish on the web, we will be enforced to process the HTML pages. Generally, the approach of processing HTML pages by HTML parse
algorithms has no efficiency. Because, parsing approaches of HTML structure includes various states. Furthermore, there are a lot of data as representing elements in HTML document which make the identification of data difficult let alone to extracting ontology of data. Nevertheless, management of the most websites by content management systems helps to discover appropriate patterns to parse HTML documents and extracting data effectively. Here, we are trying to present an approach based on surveying HTML pages for a specified domain. The approach uses facilities which a content management system provides, and illustrate a pattern to identify scheme of domain database. Then, we extract ontology domain from a created scheme. Suggested approach semi-automatically performs ontology extraction based on HTML pages of targeted domains. First, patterns available to extract data of identified domain which exist in several places of web are identified by domain expert. Then, based on the patterns related to each one of databases, information about domain is extracted from HTML pages and deal with the simulation of relational database scheme. During the next step, we extract the domain ontology from simulated database semi-automatically. Since there are natural differences in the ontology of various websites related to a specified domain, we suggest an approach to match several different ontologies in specified domain. Possibility, measurability, and extensibility are the main properties of suggested approach. In the following we will introduce the structure of CMSs and their roles in construction of ontology.

A. Extraction of relational scheme from HTML pages

We have reviewed the approaches of creating ontology from HTML pages in part 3. As it was stated there, using HTML pages can help to improve constructed ontology from relational schemes. However, using HTML documents have benefits, which ultimately can lead to improvement of extracted ontology.

Overview of tree structure of HTML is as follow:

```html
<html> <head> 
........<!— page information tags—> ........ 
</head> <body> 
........ <!— The main body tags—>........ 
</body> 
</html>
```

Each HTML file creates with `<html></html>` tag and the content are placed between the tag. HTML pages are composed of two `<head>` and `<body>` parts. Information of the page that don’t render are placed in `<body>` part. In this part, important tags such as Title, Style, Script, Nextid, Meta, Link, and Base are applied. We can consider following advantages for work with HTML pages:

- Each HTML page has a unique URL which parts of the URL states its content and makes possible the class determination of that data.

- Information is placed in a tree structure in HTML page. Therefore, information available in sub-nodes of a node is considered as part of main node concept or a subclass of it.

- Information is placed in each location of HTML page in a tag. Therefore, it can identify the existing relationship between data placed in the same tag in a set of different pages. General structure of a tag in HTML is as follow:

  `<tagname id=idname option1 =value1... option k =value k> Tag Value </tagname>`

If each tag id is determined accurately, it can simply identify the considered tag through it. `<title>` tag specifies the title of a page. `<meta>` tag is used for various applications such as determination of page language, page keywords, page distribution, and such cases. We have brought some of these applications.

1. `<meta content="fa http-equiv=Content-Language/>
2. `<meta content="text/html; charset=windows-1256" http-equiv="Content-Type"/>
3. `<meta content="semantic web laboratory " name="keywords"/>
4. `<meta content="shahid chamran university of Ahvaz" name="description">
B. Automated crawling of HTML pages based on pattern

After performing the appropriate pattern to extract web relational data scheme through HTML output by CMS, we can automatically restructure both of the complete scheme of database and its samples. The structure leads to a relational scheme-like, as the main features are identified and a relational table is created based on it. Construction of created scheme-like records is performed by a designed search engine completely automatically. In this part we will introduce the way of surveying the pages by crawlers of this search engine which support the focused search (FS). Architecture of suggested crawler has solved two common problems of navigation of web graph which are 1- coverage: navigation of the total receiving domain, and 2- novelty: identify repetitive data. However, the crawler has the ability to identify web boundary that pages of a certain language such as Farsi are identified from the other pages. Following figure reveals the overall architecture of crawler.

Each part working manner is as follow:

1) Arbiter: as manager of system this section follows these activities:
   - The range of coverage by each crawler and also performing crawling strategies such as allowable domains, determination of the number of loadable pages from each domain and such performances.
   - After loading the needed number of pages by Custome Crawler, manager of crawlers undertake its preparation for new crawling and establishing crawler for crawling through stopping them.

2) Sent commands are influenced from its inputs to other parts by Arbiter.

Costumed crawler work is to insert the loaded pages of each domain in pre-determined database. On the other hand, inputs of this part of crawler include followings.
- Number of loaded pages by each crawler
- Number of remaining pages in each crawler lineup
- Condition of each crawler activity (is there any task to run or not?)

- It might determine limitations like maximum number of pages allowed for this part. For example, for a specified domain like aut.ac.ir a creep having 500 pages is set. Also, considering the properties of each Customed Crawler which exists in the file entitled as Properties, Arbiter investigate the creeping activities. When one of the followings occurs, the crawling activity will cease.
  - Amount of downloaded files by crawler is more than considered maximum.
  - Crawler stop working (don’t download)
  - Crawler download more than 95 % of its queue

In each time, Arbiter checks the crawlers, if above conditions prevail, crawler stops. According to Properties file which specifies the crawler is active or not, it might create or run new job.

Designed crawlers are responsible for batch loading. In other words, in each cycle of crawling process each crawler loads the maximum number of possible pages with covered domain. Rules related to coverable domains are automatically regulated in each initiation of crawlers by crawlers themselves.

3) In PSRDB part, management of relational database, which information extracted from crawlers is provided for it, is performed. In other words, limitations and data filtering obtained from crawlers is managed in this part.

In the following we will deal with th way of parsing HTML pages.

C. Parse of HTML pages and construction of PSRDB
Extract information from web databases by navigation of webpages following HTML standard. HTML structure of a structure is tree and hierarchical and in fact it is a simpler type of a XML document. Following code represents the general form of HTML.

Root of tree including div tag is in the above code which it contains two properties named as id and class. Basically, amount of id property must be unique in the entire page. When crawler sends a page from user to server, it uses HttpClient component. The component provides the access to the HTML document and its components through taking that page address. In this connection, it may be difficult to access information. In the following, we will suggest an answer to solve it by stating the details.

✔ Problem of access to pages from reference link

Navigation of pages based on a preliminary list is begun as root list and go to other pager through this list and more than construction of total scheme-like structure during construction of ontology domain and sub-domains uses this connection. Sometimes access to navigator may not be possible automatically. Some of these cases are when the pages are based on Ajax or javascript pages.

1) Ajax

It cannot navigate and retrieve the HTML structure of Ajax based pages through HttpClient. In pages used Ajax to retrieve information from database program in their code, page content change through buttons without change in page address. In this approach, the information of a HTML page will not retrieve from database after request, but some parts can retrieve and represent from database through jQuery Ajax requests. For example in http://eng.ui.ac.ir/user address of University of Esfahan, as shown in figure 8-3 part of page information is called via Ajax through buttons of “articles, thesis, books and pamphlets”.

Above address has used jQuery Ajax to retrieve information from database. Figure 9-3 represents the code of this address.

The problem created by code is when a request from HttpClient is sent to receive HTML the page doesn’t retrieve. Therefore, it is necessary to crawler search information in page structure in the pages used jQuery Ajax. In other words, the search is manually done in page code and related Url in found manually, and based on the Post or Get of the request, request is sent to related webpage. For example, in above code Url is EducationHistoryPage.aspx/GetInfo and sending parameter is staffid. The problem also exists for retrieve information in other pages, as some CMSs have used an approach rather than jQuery Ajax, and it cannot automatically access to the information.

For example in the following address:


Each of the links in the right side of page is retrieved as Ajax, but it cannot find a way to retrieve information by navigation of code. Thus, although there may some modes of Ajax commands can suggest solutions, but there is no general solution for such CMSs.

2) JavaScript

Another problem in retrieving webpages is using javascript to layout different parts. For example, the page with following address is used javascript to represent different parts of publication.

publications-siansadoghi.profcms.um.ac.ir/index.php/per-http://h
Command javascript:_doPostBack('uc_articles$gv','Page$2)' perform retrieving different information for the page. According to figure 10-3 which represents the page, four different parts is provided for various publications by javascript. When user click on each button, a server side method is invokes and retrieve information from program database. Since there is no specified Post and Url in this method, it cannot retrieve information.

CONCLUSION

In this study we have investigated the extraction of ontology from web pages. Investigation of practical ontology extraction from mass HTML documents is conducted in this study. For some problems such as absence of specified word net in different domains and weak processing support of web standard for Persian language, ontology extraction had no useful results for Persian web. In this study, for the first time it is tried to develop the semantic web in Persian web domains by focusing on extraction of Persian web ontology. After studying available approaches which directly transfer the relational web database to ontology, we have divided the approaches to five groups according to input source which is relational database. Input source for database can be relational scheme, SQL-DDL physical scheme, ER/EER plot, or HTML forms. In each group, the main existing methods and their shortcomings are discussed briefly. Ultimately, we concluded that the best source of ontology production from relational database is using HTML pages when structure of database is not available. Existence of different ontologies is unavoidable due to difference at the level of ontology. Therefore, semantic web has encountered integration of ontologies to create connection between them. Using ontologies adjustment to recognize similarities and differences between two ontologies is a practical strategy to solve the heterogeneity between them. Here, basic approaches of data adjustment were introduced comprehensively, which include the approaches based on limitation, graphics, re-alignment, string approaches and using linguistic resources like WordNet and Wikipedia. It was observed that instead of using intelligent algorithms in this field by different techniques such as machine learning and neural network, but it still does not provide the required accuracy.

✔ Advantages

Suggested approach which is based on navigation of HTML pages, has numerous advantages. The most important advantage is its accessibility when database scheme is no accessible. Today, a considerable part of what is in the web as content, is saved in web databases on servers, and after taking a regeneration cycle in representable formats for users it will publish in the net. There are several databases with different structures publishing various contents and similar websites, which mostly there is no direct access to them. However, it is necessary to access these databases to create semantic web. Ontology construction and creation of metadata tags is solely possible through this way.

✔ Disadvantages

Web based approaches have disadvantages such as high computational overhead and ontology accuracy. These approaches need much research on the data extraction pattern from HTML structures, as it performs retrieving data effectively.

✔ Future works

We want to perform complete implementation steps of an ontology based on Persian domains. This requires the expansion of Persian WordNet. The complete details of the process of semi-automatically create OWL ontology corresponding to the content of relational database based on the analysis of its
related HTML-forms. Our approach can be used for migrating HTML pages (especially those that are dynamically generated from a relational database) to the ontology-based Semantic Web. The main reason for this migration is to make the relational database information that is available on the Web machine processable, and reduce the time consuming task of ontology creation. However, in the most circumstances, the obtained ontological structure is coarse. In addition, some semantics of obtained information need to be validated. So refining obtained ontological structure is necessary. Because existing repositories of lexical knowledge usually includes authoritative knowledge about some domains, we suggest as future work refining obtained ontology according to them, especially machine readable dictionaries and thesauri (e.g., WordNet).

References


