International Journal of Management, IT & Engineering
(ISSN: 2249-0558)

CONTENTS

Sr. No.	TITLE & NAME OF THE AUTHOR (S)	Page No.
1	**A Study on the Job Stress in Association with Personal Attributes of University Employees in Nepal.** Shyam Bahadur Katuwal	1-23
2	**A Comparative study of the Relationships between Multiple Intelligences and General Self Efficacy among Public and Private Organizations in Maragheh.** Gholam Reza Rahimi and Mohammad Reza Noruzi	24-39
3	**A STUDY TOWARDS OVERCOMING EMPLOYEE RESISTANCE TOWARDS TIMESHEET.** B. Koteswara Rao Naik and M. Kameshwara Rao	40-55
4	**An Integrated Cryptographic Algorithm based on Biometric Features.** S. Sathyavathi and P. Krishnakumari	56-71
5	**An Efficient Model to Improve Software Development Process and Quality Assurance.** Ajay Jangra and Sachin Gupta	72-89
6	**Reliability Prediction of Fault-Tolerant Multicomputer Interconnection Networks.** N.K. Barpanda, R.K.Dash and C.R.Tripathy	90-109
7	**The Moderating Role of Supporting Technology on the Relationship between Firm Integration and Supply Chain Orientation: An Emperical Investigation of Consumer Goods Industry in SOUTH SUMATERA INDONESIA.** Inda Sukati, Abu Bakar Abdul Hamid, Rohaizat Baharun and Huam Hon Tat	110-142
8	**Searching and Integrating Query Interfaces using Domain Ontology.** Anuradha and A.K Sharma	143-161
9	**Identification of Paraphrasing in the context of Plagiarism.** Nidhi Kushwaha, Deepak Kumar and Dr. P. R. Gupta	162-175
10	**An efficient implementation of Triple DES (Data Encryption Standard) through Hash function.** N.Venkatesan	176-200
11	**Health Education and Quality of Life: The Santal Community in Bengal.** DR. SHARMISTHA BHATTACHARJEE	201-218
12	**Police Observations of the Durable and Temporary Spatial Division of Residential Burglary.** M.Vijaya Kumar and Dr .C.Chandrasekar	219-240
13	**Frequency Control in Interconnected A.C. Systems through HVDC Link Using Artificial Intelligence.** Dr. Anil Kumar Sharma and Dr. G. K. Joshi	241-255
14	**Challenges and the Future Perspectives of labor Related Issues in Internationalization.** Sirous Fakhimi-Azar, Farhad Nezhad Haji Ali Irani and Mohammad Reza Noruzi	256-271

Chief Patron

Patron

Editorial Board

Editorial Advisory Board

Mr. P. PRABHU
Assistant Professor, Alagappa University, KARAIKUDI

Mr. MANISH KUMAR
Assistant Professor, DBIT, Deptt. Of MBA, DEHRADUN

Mrs. BABITA VERMA
Assistant Professor, Bhilai Institute Of Technology, DURG

Ms. MONIKA BHATNAGAR
Assistant Professor, Technocrat Institute of Technology, BHOPAL

Ms. SUPRIYA RAHEJA
Assistant Professor, CSE Department of ITM University, GURGAON

Title

SEARCHING AND INTEGRATING QUERY INTERFACES USING DOMAIN ONTOLOGY

Author(s)

Anuradha
Computer Engg. Department
YMCA University of Sc. & Technology
Faridabad, India

A.K Sharma
Computer Engg. Department
YMCA University of Sc. & Technology
Faridabad, India

Abstract:

In addition to billions of Web pages indexed by search engines, the Web also contains a large number of databases whose contents is 500 times larger than Surface Web and are only accessible through query interfaces. This type of web is known as Hidden Web. Its content is out of reach of traditional search engines and its growth is increasing at extraordinary rate. To search the hidden web data, one has to fill these HTML forms manually. That's why Search query interfaces are considered as entrance to Hidden web databases. So, to explore the Hidden Web, we must explore its entrance first. Many different integration solutions have been proposed so far. This paper proposes a technique to detect and construct an integrated query interface that integrates a set of web interfaces over a given domain of interest. This interface is used to fire the queries over various search interfaces of hidden web sites. The proposed strategy does that by focusing the crawl on a given topic i.e by wisely looking at domain ontology which leads to pages that contain domain specific search forms.

Keywords: *Hidden web (deep web), search query interface, Hidden web crawler, Domain Ontology.*

INTRODUCTION:

Search query interfaces act as entry points to the Hidden web which contains substantial amount of high quality structured data. This data is out of reach of traditional search engines. The traditional search engines use inverted index as a data structure to index the web data and keyword interface to retrieve the data. But, surfacing the Hidden Web is more difficult task in many respects. First, the index structures for the hidden web deal with the structured data as well as the large volume of data. Second, the Search query interfaces often have more than one attribute and require their respective values to be submitted. For data sources with multi-attribute interfaces, multiple attributes and their respective values should be stored and the resulting HTML pages should be added into a deep web search engine index.

For instance, suppose a user wants to search some information (say price of a used car) hidden behind a search interface on a site. The user needs to follow the following steps.

1. He/ She has to discover the URLs of the used car sites.
2. Visits homepages of all this kind of web sites.
3. Send queries through HTML forms.
4. Extract the relevant information from the result web pages.
5. Compare or integrate the results from multiple sources.

Two examples of search query interfaces are shown in figure 1. User has to fill the fields of these types of forms and then he/she will get the result pages.

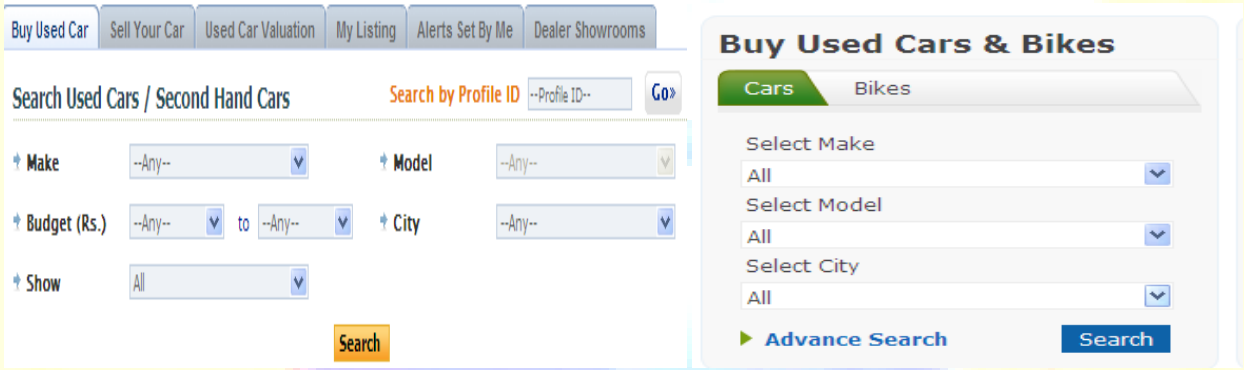

Fig 1: Search interface Examples

Therefore, there arises the need for new information services that can help users to find the information in integrated form. To minimize user effort, the problem of automatically interaction with information sources in the Hidden web is explored in this paper.

The paper has been organized as follows: Section 2 describes the proposed work that detects and extracts the search query interfaces, by looking at domain word, attributes and their synonyms in the source code of the web page. If these are present then this means this interface is search interface of our interest. The section also compares the efficiency of proposed method with the previous one [1]. Section 3 constructs an unified search interface by accumulating query interfaces to the Base interface. Section 4 draws the conclusion and describes the future research.

Proposed work:

A search query interface allows a user to search some set of items without altering them. The user enters a query, by typing or selecting options, to describe the items of interest. Results might be a page linking to items, a page containing items or a single page. The items found should match the query. The majority of search interfaces on the Web are HTML forms. Therefore the problem is: given a URL List, find all HTML search interfaces (search forms).

This proposed architecture automatically detects the domain specific search interfaces by looking the domain ontology for particular domain in the source code of web page. The framework is shown below in figure 2.

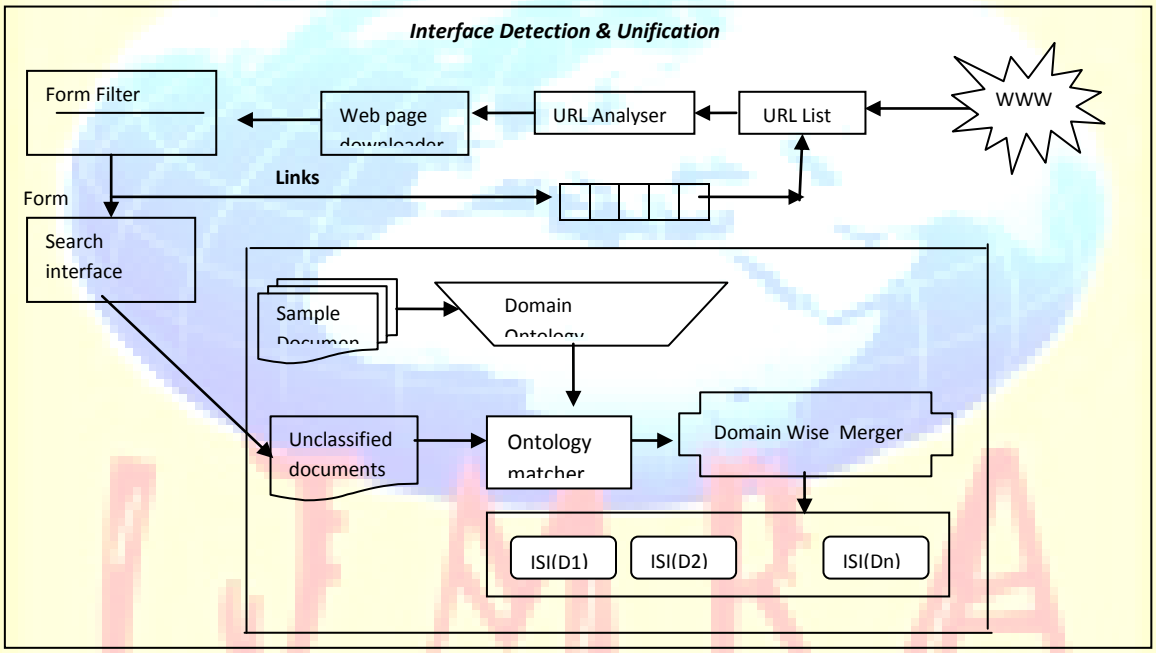

Fig 2: Interface Detection and Unification Architecture

Building Domain Ontology:

Domain ontology can be defined as information architecture in a specific domain [10]. It provides a structured way of describing knowledge. It also defines the concepts about Web page categories and their hierarchical relationships. Ontology is a specification of conceptualization that is description of concepts and their relationships[10]. Ontology is composed of concepts,

attributes and the relation among concepts. Concept is anything that can be described. It can be a real, assumed, concrete or abstract. A concept in ontology can be described by the attribute values. The main function of ontology is to provide the knowledge base needed for the classification of search results [6]. Organizing search results into hierarchical structure can help users navigate, seek and find more quickly information they are looking for.

Since Hidden Web is a huge repository for all domains, we confined our research to car domain. To build the ontology in this domain, used car keyword is pushed into the search box of Google and result index is extracted as shown in figure 3.

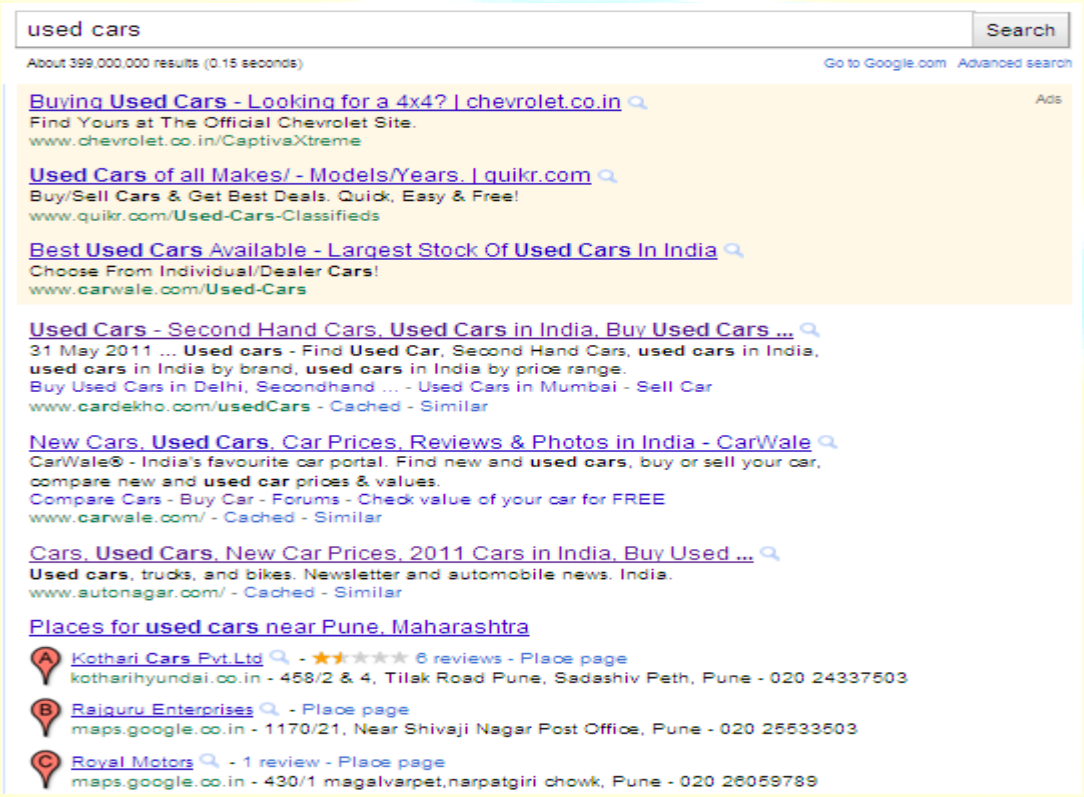

Fig 3 Google results for used cars

Attribute Extraction:

Out of large set of results, 4 sites are taken randomly as sample documents. These sites are downloaded to extract the search attributes. Algorithm for extracting the attributes in given below in figure 4.

Fig 4 : Algorithm for

Algorithm1 atrr_extract(URL List)

1. Alist=ϕ and i=0;

2. Pick URL one by one from URL List

2. x=document.getElementsByTagName("select");

// reach upto the point where select is present.

A

ttribute Extraction

This algorithm works by taking the URL one by one from the list. Because all the attributes of search form are proceeded by tag <select>, attributes can be extracted by extracting the text value (innerHTML) followed by that tag. The output of this phase is given in table 1.

Table 1 Extracted Atrribute list

Website	Attribute list
www.autonagar.com	make, model, city
www.carwale.com	make, model, city, price
www.gaadi.com	Make, model, budget, city
www.cartradeindia.com	Brand, model, state, city

Interface Detection and Extraction:

To detect and extract search query interfaces, an algorithm haven below in figure 5 has been proposed which looks for the synonyms of domain word and also the synonyms of attributes of sample form. To find the synonyms, wordnet (online dictionary) is used[12].

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonym sets[12].To make the onology, we have used this dictionary and synonyms are extracted and a list is made for each attribute. Synonyms for make are shown in figure 5.

.

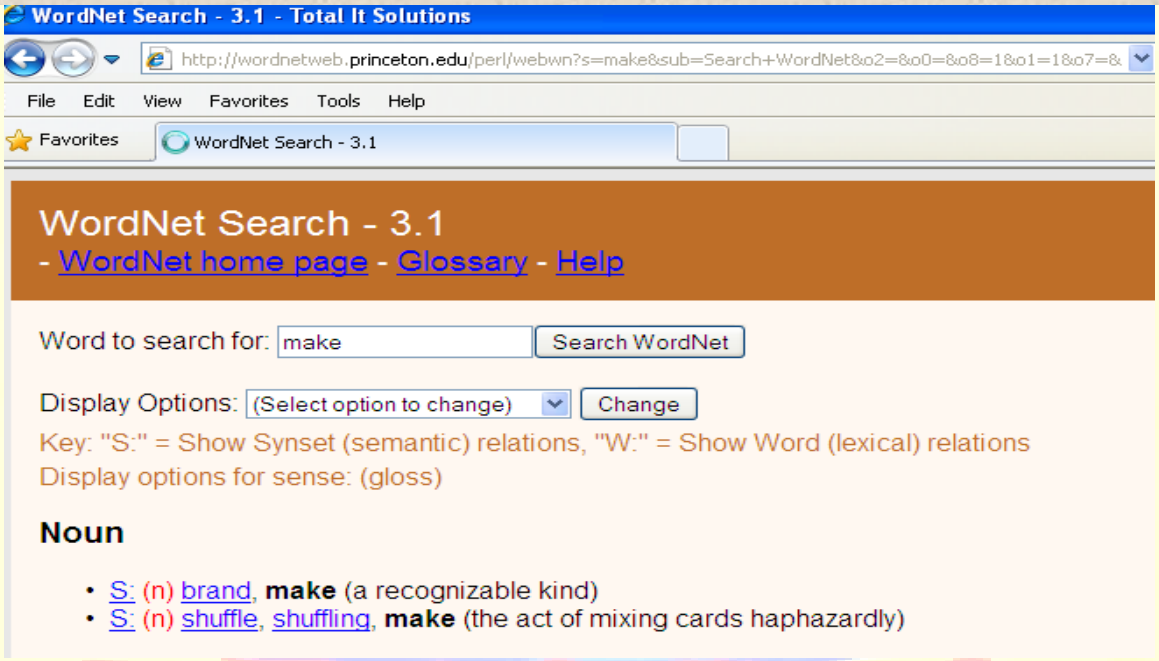

Fig 5 synonym list for make from wordnet

For make, the synonyms are (brand, shuffle) .Similarly for model, synonyms are (framework, example, role model, poser etc.), for price list is (Price, monetary value, cost) and for city (metropolis, urban centre) is the list.

Ontology Table:

To make the domain ontology repository, synonyms from the wordnet dictionary are collected along with the attributes from all 4 sites and are stored in database as shown in Table 4.2.

Table 2 Ontology table

Atrribute(ai)	Synlist
Make	Select make, brand
Model	Model, framework, example, poser
City	City, cities, metropolis
Price	Price, range, budget

Algorithm for Interface detection is given below in figure 6.

Algorithm Interf_detect2(URL List)

1. Pick one by one URL from URL list
2. Check for the string login or registration or signup in URL

   If (present) Then " it is a login/ registration form"; discard URL and Go to step 1 .

   Else Goto step3.
3. Download the source code and save.
4. check the source code for tag <form> and </form> // Form analyses

   if (present) Goto step 5

   Else " It is a simple web page". Goto step 1.
5. check source code // Search interface analysis

   if (input-type="login" or "registration" or" sign up")or (having password control)

   then Goto step 1.

   Else Goto step 6.
6. if the htmlsrc.Contains("make")|| htmlsrc.Contains("brand") && htmlsrc.Contains("model")||htmlsrc.Contains("framework")||htmlsrc.Contains("example") ||htmlsrc.Contains("models") && htmlsrc.Contains("city") || htmlsrc.Contains("cities") && htmlsrc.Contains("search") && htmlsrc.Contains("price") || htmlsrc.Contains("cost") || htmlsrc.Contains("budget") || htmlsrc.Contains("range") && htmlsrc.Contains("car"))|| htmlsrc.Contains("used car") || htmlsrc.Contains("submit") ||htmlsrc.Contains("search")

   then Goto step 7.

   Else Goto step 1.
7. Make a list of URLs i.e the list of web search interfaces

Efficiency of proposed algorithms:

There are two widely used metrics for evaluating the correctness or accuracy of algorithms. These are Precision and recall.There are four ways of being right or wrong:

1. TN / True Negative: case was negative and predicted negative
2. TP / True Positive: case was positive and predicted positive
3. FN / False Negative: case was positive but predicted negative
4. FP / False Positive: case was negative but predicted positive

Precision is defined as fraction of retrieved documents that are relevant to the search. High **precision** means that everything returned was a relevant result.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Recall is defined as fraction of documents that are relevant to the query that are successfully retrieved. In even simpler terms, a high **recall** means you haven't missed anything.

$$\textbf{Recall} = \frac{tp}{tp + fn}$$

Accuracy:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

Table 3 summarizes the efficiencies of the first method that is proposed earlier [1] and the proposed method.

Table 3: results for two algorithms

Algo	tn	Tp	Fn	fp	Precision	Recall	Accuracy
1.	9	10	1	2	83%	90%	86
2.	12	10	0	0	100%	100%	100%

Unification of Interfaces:

Structured websites tend to publish different product features and properties. For example product car has following properties (make, model, city, year, price). Website w1 want to give (make, model, city, year, price) fields in the search form, website w2 may select make, model, city only. There would not be problem with user, but problem is for someone who collects and compiles this information.

There is no proper solution for this. We can take the union of two attribute list like (make, model, city, year, price)∪(make, model, city) = (make, model, city, year, price).

For the experimentation purpose, we have chosen 4 sites and the attributes are:

Table 4 Attribute list from different websites

Website	Attribute list
www.autonagar.com	make, model, city
www.carwale.com	make, model, city, price
www.gaadi.com	Make, model, budget, city
www.cartradeindia.com	Brand, model, state, city

Unified list of attributes:

(make, model, city) ∪ (make, model, city, price) ∪ (Make, model, budget, city) ∪(Brand, model, state, city) = (make, model, city). After getting the unified list of attributes, ontology table i.e table 2 is again modified as shown in Table 5.

Table 5: Ontology table2

Atrribute(ai)	Synlist
Make	Select make, brand
Model	Model, framework, example, poser
City	City, cities, metropolis
Price	Cost, range
Year	yr

An algorithm for the method of interface unification is given below in figure 7.

Algorithm for Unification of Interfaces:

Algorithm Unif_interf (URL List)

1. For each URL in URL List U1,U2,…….Un
2. NAlist = atrr_extract(Ui);
3. For each attribute from NAlist na1,na2,...na(i), na(n) repeat step 4.
4. if na(i) = a(i) in Table 4.2
   then rename na(i) with a(i) and GOTO step 5.
   else exit;
5. Display the list of attributes and form a new search query form.

This algorithm works by taking URL one by one and extracting the attributes from all websites. If the attribute from particular website matches with the attributes from the ontology table 2 then the unified interface will have the attribute with the same name as in ontology table and if the attribute is not present then it will be added to attribute list.

Experiments and results:

For experimental purpose, we have taken 22 sites of used car domain and made a crawler in C#. This crawler returns “valid” message if the site contains search interface and displays “invalid “ message it does not. Results are shown in figure 8 and it shows 100 % results.

link	status
http://www.carwale.com/used/search.aspx	valid
http://www.quikr.com/	invalid
http://www.carsalesindia.com/buycars/usedcars/	valid
http://www.cardekho.com/used-cars/	valid
http://www.good.com/	invalid
http://www.estatesmall.com/	invalid
http://maps.google.com/	invalid
http://www.dhingramotors.com/	invalid
http://delphi.com/	invalid
http://www.cartradeindia.com/buy/usedcars	invalid
http://www.autonagar.com/	valid
http://www.usedcars.com/	valid
http://www.drcar.co.in/FeaturedUsedCars.aspx	valid
http://www.automartindia.com/	valid
http://www.carsingh.com/	valid
http://www.freeads.in/	invalid
http://www.marutitruevalue.com/buy.aspx	invalid
http://www.imdb.com/	valid
http://www.wikipedia.org/	invalid
http://www.jmdcars.com/usedcars.asp	valid
http://www.jaade.com.tr/	invalid
http://www.cars24.com/	invalid
http://www.car200.com/	invalid

Fig 8: Results of Proposed system

Conclusion:

This paper automatically detects the domain specific search interfaces by looking the domain word, its synonyms , attributes from sample documents and its synonyms from wordnet in the source code of webpage. The problem of integrating large-scale collections of query interfaces of the same domain has been designed and developed by transforming a set of interfaces in the same domain of interest into a global interface. This interface will permit the crawler to fire the

queries to interfaces of target sites and it also provides users to access information uniformly from multiple sources of a given domain.

References:

- Anuradha, A.K.Sharma, "A Novel Approach for Automatic Detection and Unification of Web Search Query Interfaces using Domain Ontology" International Journal of Information Technology and knowledge management(IJITKM) July-december2010, Vol.2 No.2.pp 196-199
- BrightPlanet Corp. "The deep web: surfacing hidden value."
- S. Lawrence and C. L. Giles. Searching theWorldWideWeb. Science, 280(5360):98–100, 1998.
- Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to Map between Ontologies on the Semantic Web. In WWW, 2002.
- H. He, W. Meng, C. Yu, and Z. Wu. WISE-integrator: An automatic integrator of Web search interfaces for e-commerce. In VLDB, 2003.
- Masayu Leylia Khodra1*, Dwi Hendratmo Widyantoro, "An Efficient and Effective Algorithm for Hierarchical classificationof Search Results ", Proceedings of the International Conference on Electrical Engineering and Informatics Institute Teknologi Bandung, Indonesia June 17-19, 2007
- S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In Proceedings of VLDB, pages 129–138, 2001.
- B. He and K. C.-C. Chang. Statistical schema matching across web query interfaces. In Proceedings of SIGMOD, pages 217–228, 2003.
- Wright, Alex (2009-02-22). "Exploring a 'Deep Web' That Google Can't Grasp". New York Times.
- T.R. Gruber. A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2):199-220.

- Gruber, T. R. (1993). Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands. Kluwer AcademicPublishers.*
- http://wordnetweb.princeton.edu/perl/webwn