International Journal of Grid and Distributed Computing
Vol. 9, No. 6 (2016), pp.381-392
http://dx.doi.org/10.14257/ijgdc.2016.9.6.33

Automated Data Extraction with Multiple Ontologies

Jer Lang Hong

School of Computing and IT, Taylor's University, Malaysia
jerlang.hong@taylors.edu.my

Current search engines require an accurate yet fast automated extractor to extract relevant information from deep web for the users. Human users usually enter search queries and the search engines will then locate the desire information of interest by disambiguate the search query accordingly. The queries will then be passed on to multiple search engines for further processing. These search engines will then return the search results to the main search engine. However, data returned from these search engines are usually varied and presented in numerous formats and layouts. To extract them, we need automated extractor to filter out irrelevant information and locate the correct information. Current trends focused on using ontologies to automatically extract this information with high accuracy. To the best of our knowledge, no works have been made on using multiple ontologies (using many ontology techniques) to automatically extract information from deep webs. In this paper, we demonstrate that multiple ontologies technique can achieve higher accuracy when extracting data from the deep web. Our method outperforms existing state of the art systems and is able to robustly extract data from deep web.

Keywords*: Automatic wrapper, deep web, search engine*

1. Introduction

The evolution of World Wide Web has seen a dramatic increase in the number of web pages. Statistics shown in http://www.worldwidewebsize.com/ indicates that there are billion of web sites available currently. With the evolution of Web 2.0, web pages have been dynamically generated using server side scripts, and with the recent trends of social networking sites and microblogging, it is very easy to generate websites of interest for the users. The early days of having the server to host and display the webpages is no longer applicable, users can easily generate webpages by putting comments and posts on the web.

To facilitate browsing and locating these web pages, search engines are developed where they can locate these web pages through web crawlers, extract and index them accordingly (Figure 1, Figure 2, Figure 3). However, extracting and indexing these webpages is a non trivial task, as webpages are usually formatted in different layouts. Moreover, HTML language is ambiguous and lack uniformity in its design.

Many approaches have been proposed to resolve this issue. The first approach is to utilize the underlying coding of the HTML page, which is the DOM Tree (Figure 4). Various properties of DOM Tree have been utilized, such as their location and hierarchy within the DOM Tree. The second approach is to use the visual properties of data, such as visual boundary, text color, and size.

ISSN: 2005-4262 IJGDC

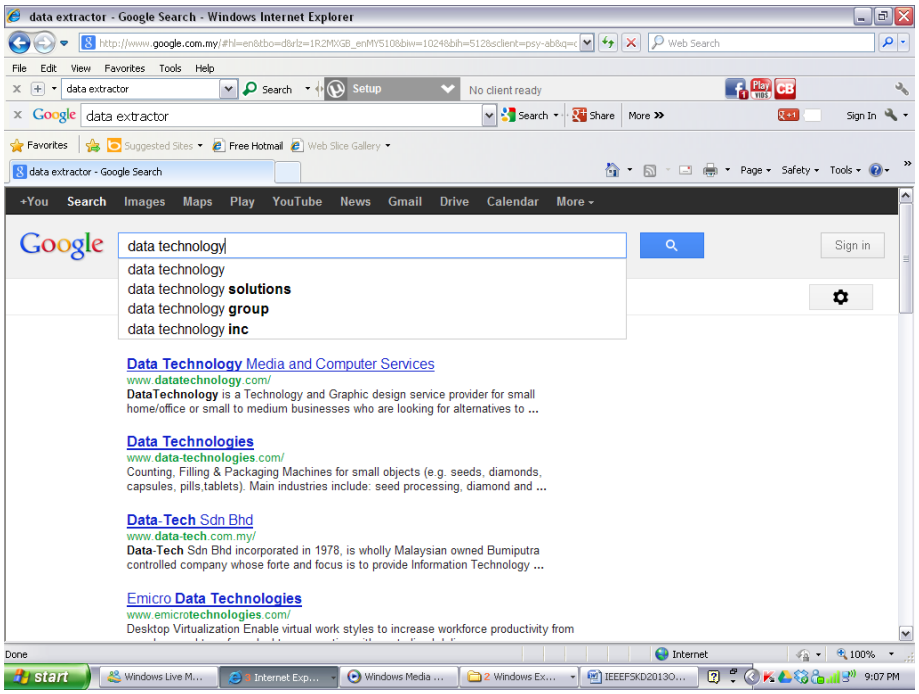

Figure 1. Google Search Engine

Both approaches are efficient, but they are not without problems. The first approach, though they are fast in general, is not generally applicable for most of the cases in data extraction. It fails to consider for many factors, the most common ones are disjunctive and optional data. The second approach is generally more accurate than the first one, as it considers for human visual perception on the webpage. However, this technique is generally slower than the first approach as it needs more processing to obtain the visual information from the underlying browser rendering engine.

Recently, works have been carried out in using ontological technique for extracting data from deep web. This technique has proved to obtain better accuracy than the DOM Tree and Visual Cue approaches as they make use of additional level of information compared to the previous two. However, we are of the opinion that if multiple ontological techniques are to be integrated into one single coherent approach, there will be many more extra layer of information available for extraction to be carried out. This could lead to higher accuracy in data extraction. To the best of our knowledge, there has not been any work for data extraction using multiple ontologies technique.

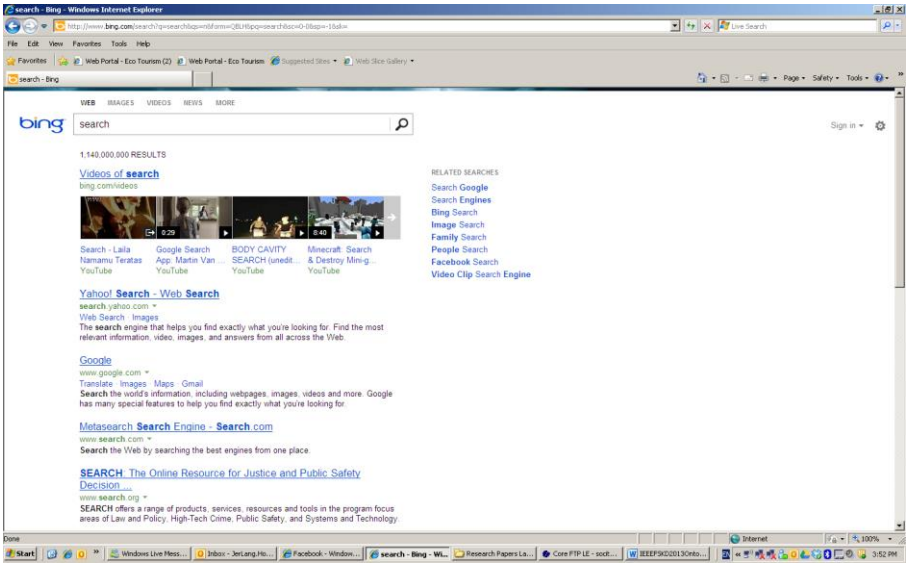

Figure 2. Bing Search Engine

Figure 3. Yahoo Search Engine

In this paper, we propose an extraction module incorporating multiple ontologies techniques. We use three state of the art ontologies in our approach, WordNet, CYC, and Wikitology. WordNet is chosen to identify the synsets and word similarity of data. CYC is used to identify the semantic relatedness of one data with the other while Wikitology is used to identify the relevance of each data in the webpage. These thee ontology tools are chosen due to the wide recognition they have obtained in determining the semantic properties of data. In fact, W3C committee has widely endorsed these techniques as state of the art. For example, CYC ontology has been widely recognized for its ability to detect counter intelligence information. Wikitology on the other hand, its well known for its data representation, particularly the semantic web it has provided to link many concepts and terms together (Figure 5). Lastly, WordNet is known for its ability to check the relation between taxonomies and terms. With its huge database, it is able to represent almost all the concepts available in this world. Recently, WordNet has provided support for a variety of languages, and it is able to run across many different platforms. From our observation,

deep web usually contain data which are related semantically, in terms on their synonymity, topics, events, and relationship.

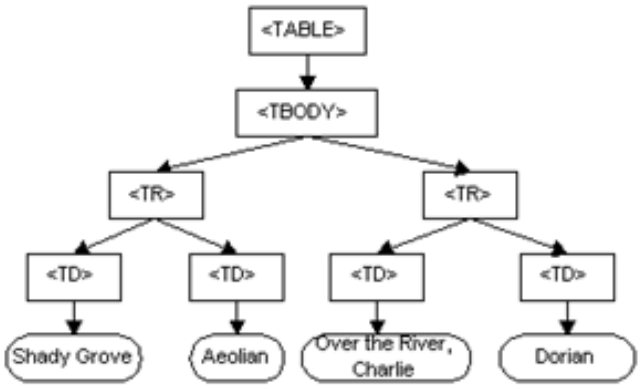

Figure 4. Document Object Model (DOM) Tree

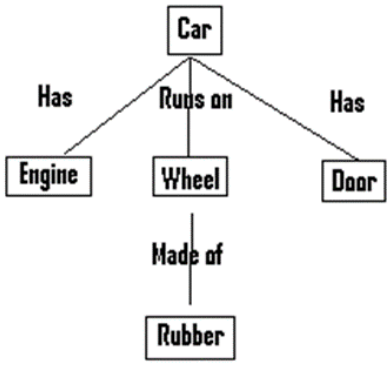

Figure 5. Semantic Web

This paper contains several sections. Section 2 describes the current work that is related to ours. Section 3 provides the methodological approach of our method using Ontologies. In Section 4, we demonstrate experimental tests conducted on our method. Finally, Section 5 summarizes our work.

2. Related Work

2.1 Current Extraction Tools

Currently, there are three approaches to extract data from the deep webs. The first approach uses DOM Tree properties such as parent child relationship, hierarchical structure and the sibling nodes in order to determine the relevant data. MDR [2] uses Generalized Nodes to extract data, assuming that data are ordered in repetitive mode, containing similar nodes. Similar data are grouped into similar region, forming data region. In 2005, Zhai et al developed DEPTA [26] which uses tree matching to match the tree structures of data, assuming that data not only occur in repetitive order, but they also contain similar tree structures. Other approaches that use tree matching algorithms are such as NET [26] (using nested tree matching), WISH [13], [14], [15] (using tag counting). A variant of tree matching is developed in 2005, which uses primitive tandem repeat to match data with similar structures.

The second approach uses visual cue to match data in deep webs. The most common ones are the work of ViNT [11], where data are separated into content lines, and then grouped into blocks forming data records. Blocks of data records can then be further grouped into region. VSDR [] uses the visual boundary of data region to extract data from deep webs, where it assumes that data region which is centrally located and the largest are the relevant ones. ViPER [17] uses the visual boundary of data records where it assumes that almost all data records, if not all, contains similar visual boundaries.

2.2 Ontological based Extractors

ODE [23] is the first ontological based wrapper to utilize ontology to extract data from the deep web. It uses entropy probabilistic model to determine the semantic relatedness of relevant data. Due to the fact that ODE utilize ontology technique in its operation, it is the first wrapper which is able to extract single record data from the deep web, as this wrapper learn from its training data, and analyze the semantic properties of data instead of their patterns for data extraction. In 2011, Hong et al developed OW wrapper to extract and align data [14]. WordNet is used to analyze the semantic properties of data, where it checks for synonymous words and word disambiguation. OW wrapper is able to accurately extract data from the deep web. Other approaches that use Ontology are such as DeepMiner [27], OWL [24], and the work of Embley [5].

2.3 State of the art Ontology Tools

2.3.1 WordNet

WordNet [3], [22], [24] was developed in 1998 as a light weight ontological technique, closer to a thesauri, and it is a lexical database for English for the semantic matching of words in Information Retrieval research [1], [4], [9], [10], [12], [20], [21]. WordNet contains a huge amount of information (150,000 words organized in over 115,000 synsets for a total of 207,000 word-sense pairs). WordNet represents nouns, adverbs, verbs and adjectives as a group of cognitive synonyms (synsets) with their own distinct concepts. Synsets are linked by means of conceptual semantic and lexical relations. A browser is used to manage and navigate the individual component in WordNet. It categorizes English words into several groups, such as hypernyms, synonyms, and antonyms.

2.3.2 CYC

CYC is developed by Lenat [8] as part of his research work for MCC Corporation. Unlike WordNet, CYC covers a larger domain and provides more semantic information to the users. CYC provides more than hundreds thousands of terms, and millions of assertions related to the terms. The ontology in CYC knowledge has 47,000 concepts and 306,000 facts browsable by CYC web interface. CYC uses a mapping to define the concepts of each term. For example, CYC provides part of relationship between tree and leaves (leaves are part of a tree). Every concept mapped to the terms will return either a true or false statement. Based on this return value, users can then decide the appropriate actions for future processing. CYC has been successfully applied to Terrorism Knowledge Based application and has been used as part of Cyclopedia database (combining info taken from Wikipedia). However, studies indicate that CYC system and its underlying database is complicated, and it is also not scalable to large systems.

2.3.3 BabelNet

BabelNet [19] is developed to overcome the drawback of WordNet. As stated in the literature of BabelNet, WordNet is a light weight ontological technique with limiting

ontology domain and capability to provide sufficient information to the users. Using the combination of WordNet and Wikipedia, BabelNet integrates the domain and knowledge base of these two systems, and could sufficiently provide the users with higher level ontology domain. In addition, BabelNet is also able to distinguish word sense disambiguation accurately using the information provided by Wikipedia domain knowledge.

2.3.4 YAGO

Yet Another Great Ontology (YAGO) is developed by Fabian and it is a lightweight ontology with extensible functionalities for high data coverage and accuracy [6], [30]. YAGO achieved an accuracy of 95% on its test cases. YAGO extracted data from Wikipedia and unified it with WordNet, and provides the users with 1 million entities and 5 million facts. YAGO also includes functionalities such as IsA as well as non taxonomical relations between entities.

2.3.5 WordNet++

WordNet++ is an extension of WordNet to solve word disambiguation problems. It extends the existing WordNet by providing extra high quality information from Wikipedia. WordNet++ could give high quality semantic information to the users, with support for word disambiguation using the interface of supervised tool Word Sense Disambiguation (WSD) [29].

2.3.6 Wikitology

Wikitology is an ontology tool developed based on the Wikipedia. It is useful as a tool for many language processing tasks. Each article is a concept in the ontology. The terms in the article are linked to each other and they may also interlink to other documents. Wikipedia ontology is created and maintained by diverse community. It has broad coverage, multilingual, and its content is very current. In fact, the quality of its content is very high, which is useful for many research works as it is maintained and created by trusted communities.

3. Proposed Methodology

3.1 Overview

Our extractor tool is divided into four main components. We utilize three state of the art ontological tools (WordNet, CYC, Wikitology) as part of our extraction tool, which will be very helpful in extracting data from the deep web with high accuracy. The first component will separate and segment the page into several regions of interest, while the remaining three components involve determining the relevant region from the irrelevant ones. The remaining sections describe the components of our extractor tool in details.

3.2 Region Detection

We use ICE Browser to parse the webpage and have it represented in DOM Tree. Then, we use the visual cue provided by the underlying browser rendering engine to obtain visual properties of the page. To parse through the web pages, our parser needs to identify two types of objects, they are HTML Tag and HTML Text. HTML Tag is element that starts with ‘<’ character and ends with ‘>’ character. There are two types of HTML Tags, they are the opening and closing tags. HTML Text on the other hand, contains contextual

information of the webpage. Every HTML Tag usually starts with an open tag, and ends with a closing tag. In some cases, HTML Tag can both exists as an opening and closing tag. A Dom Tree is a tree containing HTML Tag and HTML Text, where HTML Text is usually the leaf nodes while the parent nodes are HTML Tags.

Our extractor will then traverses through the DOM Tree and locate block tags such as table, div. Once block tags are identified, we will then determine the visual boundaries of each tag. If the visual boundaries are within acceptable threshold, we will take these tags as regions and they will be divided according to their visual boundaries. For the case of nested block tags, we perform the previous steps recursively until the smallest block is found. Each of these regions will then go through the remaining stages described in the next section, where only one region will be identified as the correct region (the region containing data relevant to the search engine query). Unlike existing ontological technique, our approach integrates three ontologies for data extraction. We believe this approach is an advantage as it helps to provide more informative semantic properties in determining the correct data region from the incorrect ones.

3.3 Extraction using WordNet

Once all the regions are identified, we use WordNet to determine the semantic properties of the content. We use the algorithm of Jiang and Conrath [12] to check for word similarity as study in [1] shows that this algorithm yield better results than other similar algorithms. Two words are considered semantically similar if their similarity score exceeds 0.7. We match every word in the region to determine their semantic properties. Before the matching process is carried out, we remove all the irrelevant words such as punctuation, and stop words. Every word is stemmed to their base. Before the word is stemmed, we use JOrtho Spell Checker to correct the words. Once a matching occurs, the two words that are matched is taken out from the content and put into the list of similar keywords. Then, matching is carried out by matching the remaining keywords with that in the list. This process is carried out to speed up the running time for matching.

We also considered Word Disambiguation for our approach. For example, the word “interest” for the sentences “Interest in Bank” and “Interest in book” are both dissimilar despite having similar word. To differentiate this, we use Adapted Lesk algorithm [21] where the neighboring words are further examined to determine the semantic relatedness of the region. After the matching is carried out, we then examined the list of similar keywords and determine the pattern of similar keywords in the region. Regions which have highly similar keywords are identified as relevant region, hence chosen for the subsequent steps of checking. Otherwise, they are considered as irrelevant and removed from the list of potential regions.

3.4 Extraction using CYC

Once the similarity of keywords are identified and stored in a list, we will then use CYC to further checked the semantic of the keywords so that the accuracy of our system can be improved. We use CYC to relate all the keywords and from these relations, we can then further use the information provided to deduce the semantic properties. For example, the keyword Toyota can be linked to “Japanese Car Company”, “VVTi Engine”, “Fuel Efficient Car”, and “Economical and affordable car”. Using this information, we can further examine the semantic relatedness of the region. The word Toyota may not matched any of the keywords in the search results, but the extra word Car and Fuel Efficient may possibly matched the search results returned by the search engines. In addition to that, any search results returned by the search engines which are indirectly related to Toyota (e.g. Nissan fuel efficient car) may also possibly matched the information returned by CYC.

3.5 Extraction using Wikitology

From the information provided from CYC we will then use this information in Wikitology and then determined how much is the information is related to each other. For example, the search query on "Toyota" will return search results about Toyota Car Company, further information provided by CYC will reveal more information about Toyota Car. A thorough check on Wikitology will help us identify that all this information are very much related to each other. We use the API provided to plug the libraries and functions so that the matching can be carried out to verify facts returned by the search engine results. Based on all this information, we formulate a scoring function to identify how much similar is the information in the search results. From the number of links detected by Wikitology, if a link identified by Wikitology match the search results, we treat the link as correct. Otherwise, they are treated as incorrect. The percentage of semantic similarity of the search results can then be calculated as Correct links identified/Total number of links. The functionalities provided by Wikitology also help us to identify false positive information. For example, the word Windows may refer to the Operating System based Company Microsoft or simply the components of a house. Likewise the word Apple may refer to the Company Apple or just simply the type of fruit with red skin color and yellow flesh.

3.6 Filtering of Regions

We apply the semantic similarity measures to all the regions detected. Regions with semantic similarity less than 85% are discarded. Semantic similarity is calculated from the previous three steps. If a region failed to pass the test in the first step, it is automatically discarded for consideration in the subsequent steps. Likewise for the second step. If there are more than one regions remaining in the third steps, we apply heuristic technique to select one final region for our work. From the remaining regions, we apply a simple heuristic whereby centrally located region is chosen as the correct region (search results).

4. Experimental Test

We collected a sample test case of 300 web pages from the repository www.completeplanet.com. The test case samples are collected in such a way that it is distributed across a wide range of domain, such as commercial, governmental, news, and blogs. Due consideration is taken into selecting the web pages such that no duplicates web pages are selected. The sample pages are divided into three separate groups of 100 web pages each. The first sample pages group consists of randomly generated web pages. The second group consists of webpages written in various languages. The third group contains web pages with complicated layout and format. The samples from the first group are taken to test how well our approach works on real life sample pages. The second samples group is used to test the robustness of our approach in handling multiple languages web pages. The third samples are used to test the robustness of our approach in handling webpages with complicated layout and forms, particularly those web pages with highly irregular layout and not well formed. We measure the performance of our system based on two factors, precision and recall rates, which are calculated as follow:

Recall=Correct/Actual*100

Precision=Correct/Extracted*100

Where actual is the actual number of records, extracted is the number of extracted records from deep web, and correct is the actual number of records considered as correctly extracted.

Table 1. Experimental Results (Random Samples)

Terms	OntoExtract	OW [14]
Actual	1577	1577
Extracted	1598	1614
Correct	1541	1218
Recall	96.43%	75.46%
Precision	97.71%	77.24%

Table 2. Experimental Results (Multi language samples)

Terms	OntoExtract	OW [14]
Actual	993	993
Extracted	1124	1292
Correct	823	714
Recall	88.34%	76.85%
Precision	82.88%	71.90%

Table 3. Experimental Results (Complicated samples)

Terms	OntoExtract	OW [14]
Actual	1187	1187
Extracted	1224	1326
Correct	982	804
Recall	96.98%	89.52%
Precision	82.73%	67.73%

We compare our approach with state of the art Ontological Extractor, OW [14]. As shown in Table 1, our extractor outperforms OW in both recall and precision rates. This is due to the fact that our extractor utilized fully the semantic properties of data in deep web. Besides considering the word similarity, we also analyze the semantic relatedness of contents and also the relationship between each of the keywords in the data. Data are considered valid and relevant only when they are semantically similar and related, an assumption that OW did not make. When deep web contains data related to the search engine queries, most of the records in the deep web are usually semantically related, as they are search query dependent.

Table 2 shows the results of our wrapper with respect to OW when tested on multi language webpages. Test results shown that our approach works fairly well on multi language sample pages. This is due to the fact that our approach utilizes DOM Parsing and Tree Construction as part of its operation despite the fact that our ontology approach only utilize single language mode. The fact that our approach also utilizes DOM Tree as part of the extraction process indicates that our approach is well suited for real life scenario where webpages are not only written in HTML Codes, but they are also presented in numerous other languages.

Finally, Table 3 shows the test results of our approach tested on complicated web pages. These web pages contain complicated layout and structure, many of them are not well formed and valid web pages, making them hard to parse and analyze. Test results show that our approach works fairly well on complicated web pages. As long as the DOM

Parser could effectively parse through the sample pages and construct the appropriate DOM Tree, our approach is able to extract the data regardless of the layout presented in the webpages. This is because our approach analyzes the content of the webpages instead of the structure and layout, making it robust and effective in data extraction.

5. Conclusion

Extracting data from deep web is a non trivial task as it helps search engines to locate information effectively and accurately. However, current approaches have failed to extract data accurately as they use DOM Tree and visual cue for their extraction. In this paper, we introduce an extra level in our approach, which is to use ontology to extract data from deep web. We extract data by exploiting the semantic properties of data in search engine results pages. Unlike existing approaches which use ontology technique, we incorporate up to three state of the art onlotogies techniques for our extraction module. Having multiple ontologies as part of our extraction modules is an added advantage as we are able to capture more information related to the data. Experimental results show that our approach could outperform existing state of the art systems in extracting data from the deep web.

Acknowledgement

This work is carried out within the framework of a research grant funded by Taylor's University Research Grant Scheme (Project Code: TRGS/2/2012/SOCIT/022).

References

[1] Alexander Budanitsky and Graeme Hirst, "Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures," in Proceedings of the NAACL 2001.
[2] Bing Liu, Robert Grossman, and Yanhong Zhai, "Mining data records in Web pages," ACM SIGKDD, 2003
[3] Christiane Fellbaum, "WordNet: An Electronic Lexical Database," The MIT Press, Cambridge, MA, 1998.
[4] Claudia Leacock and Chodorow Martin, "Combining local context and WordNet similarity for word sense identification," The MIT Press, Cambridge, MA, 1998.
[5] D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, and D.W. Lonsdale, "Conceptual-model-based data extraction from multiple-record Web pages," DKE, 1999.
[6] Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum "Yago - A Core of Semantic Knowledge", 16th international World Wide Web conference, WWW 2007
[7] Gengxin Miao, Junichi Tatemura, Wang-Pin Hsiung, Arsany Sawires, and Louise E. Moser, "Extracting Data Records from the Web Using Tag Path Clustering," ACM WWW, 2009
[8] Guha, R.V., Lenat, D.B., Building Large Knowledge Based Systems Reading, Massachusetts: Addison Wesley, 1990.
[9] Graeme Hirst and David St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," The MIT Press, Cambridge, MA, 1998.
[10] Hicham Snoussi, Laurent Magnin, and Jian-Yun Nie, "Heterogeneous Web Data Extraction using Ontology," Proc. Agent-Oriented Information Systems, 2001.
[11] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu, "Fully automatic wrapper generation for search engines," ACM WWW, 2005
[12] Jay J. Jiang and David W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," Proc. of International Conference on Research in Computational Linguistics, 1997.
[13] Jer Lang Hong, "Deep Web Data Extraction," IEEE SMC, 2010
[14] Jer Lang Hong, "Data Extraction for Deep Web using WordNet," IEEE TSMC, 2011
[15] Jer Lang Hong, Eugene Siew, Simon Egerton, "Aligning Data Records Using WordNet", IEEE CRD, 2010
[16] Jiying Wang and Frederick H. Lochovsky, "Data extraction and label assignment for web databases," ACM WWW, 2003

[17] Kai Simon and Georg Lausen, "ViPER: augmenting automatic information extraction with visual perceptions," ACM CIKM, 2005.
[18] Niles, I., and Pease, A. 2001. Towards a Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.
[19] Roberto Navigli and Simone Paolo Ponzetto, BabelNet: Building a very large multilingual semantic network In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11-16 July 2010, pp. 216-225.
[20] Rodriguez M. and Egenhofer M., "Determining Semantic Similarity Among Entity Classes from Different Ontologies," IEEE TKDE, 2003.
[21] S Banerjee, Extended Gloss Overlaps as a Measure of Semantic Relatedness, ACM IJCAI, 2003
[22] Vossen, P., N. Calzolari, G. Adriaens, A. Sanfilippo, Y. Wilks (eds.) 1997 Proceedings of the ACL/EACL-97 workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, Madrid, July 12th, 1997.
[23] Wei Liu , Xiaofeng Meng, and Weiyi Meng, "ViDE: A Vision-based Approach for Deep Web Data Extraction," IEEE TKDE, 2009.
[24] Weifeng Su, Jiying Wang , and Frederick H. Lochovsky, "ODE: Ontology-assisted Data Extraction," ACM TODS, 2009.
[25] Wensheng Wu, AnHai Doan, Clement Yu, and Weiyi Meng, "Bootstrapping Domain Ontology for Semantic Web Services from Source Web Sites," Proc. of the VLDB Workshop, 2005.
[26] Yanhong Zhai and Bing Liu, "Web data extraction based on partial tree alignment," ACM WWW 2005.
[27] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, and Yu C., "Annotating Structured Data of the Deep Web," IEEE ICDE, 2007
[28] http://www.cyc.com
[29] http://www.loa-cnr.it/DOLCE.html
[30] http://www.ontologyportal.org/
[31] http://www.illc.uva.nl/EuroWordNet/

