International Journal of Fuzzy Logic and Intelligent Systems, vol. 11, no. 3, September 2011, pp. 171–177
http:// dx.doi.org/10.5391/IJFIS.2011.11.3.171

Incorporation of Fuzzy Theory with Heavyweight Ontology and Its Application on Vague Information Retrieval for Decision Making

Ahmad C. Bukhari and Yong-Gi Kim*

Department of Computer Science, Research Institute of Computer Science and Information Communication, Gyeongsang National University Jinju, Kyungnam 660-701, Republic of Korea

Abstract

The decision making process is based on accurate and timely available information. To obtain precise information from the internet is becoming more difficult due to the continuous increase in vagueness and uncertainty from online information resources. This also poses a problem for blind people who desire the full use from online resources available to other users for decision making in their daily life. Ontology is considered as one of the emerging technology of knowledge representation and information sharing today. Fuzzy logic is a very popular technique of artificial intelligence which deals with imprecision and uncertainty. The classical ontology can deal ideally with crisp data but cannot give sufficient support to handle the imprecise data or information. In this paper, we incorporate fuzzy logic with heavyweight ontology to solve the imprecise information extraction problem from heterogeneous misty sources. Fuzzy ontology consists of fuzzy rules, fuzzy classes and their properties with axioms. We use *Fuzzy OWL* plug-in of Protégé to model the fuzzy ontology. A prototype is developed which is based on OWL-2 (Web Ontology Language-2), PAL (Protégé Axiom Language), and fuzzy logic in order to examine the effectiveness of the proposed system.

Keywords : Fuzzy Ontology, Precise Information Extraction, Modeling Heavyweight Ontology with fuzzy logic, Imprecise Information Extraction

1. Introduction

Right decision is direct proportional to the information which is provided at the time of decision making. The increasing dynamicity in the web applications and continuous addition in web data are making the process of precise information extraction difficult. While real time applications are based on fuzzy data which does not have crisp or clear-cut boundaries. Several search engines and information extraction utilities are available on internet but for the most part, these tools are keyword base and cannot fulfill the requirements of modern era. The invention of the internet has profoundly changed our means of communication and information sharing. Easy access to the internet has played a major role in both its own growth and its utility in day to day life [1]. When the internet was invented, its vision was to create a virtual world in which human and computer work together while sharing information. Although not explicitly stated, this vision of the internet implied the development of computers with the same ability to understand content as a human. Its continuously unchecked growth derailed its structure from its vision. Due to its gigantic scale and the human ability to process information and make connections, there was not sufficient impetus to correct the deficiencies in the hardware and software side of the equation regarding relevant and correct information retrieval. Currently one has to work a lot to find correct and relevant information in a short span of time because currently the web is designed for human consumption and does not provide any help for machine interpretation. Several techniques were used in the past to make the computers operate intelligently while dealing with information. In [2, 3], web content mining and wrapper induction techniques were discussed to extract and integrate the useful data, information and knowledge from web pages. The two methods most widely used in web content mining are wrapper induction information extraction and automatic information extraction. In wrapper induction technique [3], we use machine learning techniques to make rules for information extraction. But to write wrapper classes and keep them updated is a very tough job when you consider the dynamic growth on the internet. Whenever the website contents and structure change, we have to relearn our wrapper and make significant changes according to new structures. The probability of errors occurring in wrappers is also very high. Several software were developed for visually impaired internet users to use for retrieving precise information quickly, but most of them used screen reading techniques such as *BrookesTalk* [4]. *BrookesTalk* is one of the tools which give a summary of the page to the user vocally. Another approach names as *NavAccess* is introduced in [5], which aims to facilitate the visually impaired users in website searching and navigation. In [6] the researcher presented the idea of a semantic web. The basic theme behind a semantic web is to make the web intelligent so that it can understand and conceive the meaning

Manuscript received Apr. 19, 2011; revised Sep. 15, 2011; accepted Sep. 15, 2011.
*corresponding author
This work was supported by the Korea National Research Foundation (NRF) Grant funded by the Korean Government (No. 2010-0002346).

of data [1] [7]. Ontology is one of major technique which works under the umbrella of semantic web for information sharing and reuse. The classical ontology is based on crisp methodology and cannot handle the imprecise information at its full manifestation. One possible solution to cope the issue of uncertainty is to incorporate the fuzzy set theory into classical ontology. In [8] Jun et al. presented a framework to retrieve information for supply chain management using fuzzy ontology. Sun et al. in [9] showed his work about fuzzy ontology and its usage in traditional Chinese medicine representation. The paper is discussed a fuzzy representation model along with reasoning based on fuzzy ontology. Several researcher worked on fuzzy ontology and its usage, interested can read more detail in [10, 11]. Our Proposed solution is based on heavyweight ontology and its incorporation with fuzzy set theory. Heavyweight ontology has deeper expressivity level comparing with classical ontology. The axioms increase the expression power of ontology. So our proposed heavyweight fuzzy ontology architecture (see Fig.5) is powerful enough to assist the visually impaired people to extract information from web using vocal command system and can help them in decision making process to provide right information at right time. The rest of the research work is divided in such fashion. In section 2 we explored about basic concept definitions. Section 3 is provided the way for practical modeling of heavyweight fuzzy ontology. We introduced our proposed architecture in section 4 and Section 5 belongs to experiments and results. In last section we gave concluding remarks and suggested about possible future work.

2. Fuzzy Heavyweight Ontology Concept and Modeling

Fuzzy set theory was introduced by Lotfi Zadeh in 1965[11] to deal with vague and imprecise concepts. In classical set theory elements either belongs to a set or they do not belong to a set. There is not semi participation concept in classical set theory. However in fuzzy set theory the association of an element with a particular set lies between *0* and *1* which is called its degree of association or membership degree. It adds generalization concept to the classical set theory and makes it possible to represent imprecise boundaries like hot, tall, low speed etc. A fuzzy set can be defined as:

Definition 1: A fuzzy set 's' over universe of discourse 'X' is defined by its membership function μ_s which maps element 'x' to a number between [0,1]

$$\mu_s(x): \quad X \rightarrow [0,1] \qquad (1)$$

Here $x \in X$ and $\mu_s(x)$ provides the degree of membership by which x belongs to X.

Ontology is considered a branch of metaphysics which focuses on the study of existence and an explicit specification of shared conceptualization [12]. In ontology we basically focus on concepts (classes) of the domain, their relationships and their properties. Ontology arranges the classes in hierarchical structure, in the form of sub class and super class hierarchy, and also defines which property has constraints on its values.

Definition 2: A fuzzy heavyweight ontology F_O is the combination of concepts *C*, properties of sets *Cp*, values of concepts *Cv*, relationships among concepts *Rc*, axioms *ä*. Mathematically we can defined as:

$$F_{O=}\{C, Cp, Cv, Rc, ä\} \qquad (2)$$

Where concepts, their properties and their values are all fuzzy linguistic concepts and their values fall between [0, 1] interval.

2.1 Heavyweight Fuzzy Ontology Modeling

Ontology is written in a specific language called OWL (Web Ontology language) which is developed by W3C. We used OWL-2 in our experiment which is enhanced version of OWL. Fig. 1 is showing a code view of crisp lightweight ontology class, we modeled lightweight ontology first, subsequently; we added axioms to achieve deeper expressivity level. Furthermore we exported our designed ontology to protégé *FuzzyOWL* tab and added the fuzzy concepts.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!-- Electroc Store Heavyweight Ontology Class File:AI LAB GNU-->
]>
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#"
    xml:base="http://www.semanticweb.org/ontologies/2010/11/E-Store.owl"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
            <!--
http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#accessBookAndMagazine -->
   <owl:ObjectProperty rdf:about="#accessBookAndMagazine">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <rdfs:range rdf:resource="#Book_And_Magazine"/>
      <rdfs:domain rdf:resource="#Domain_User"/>
      <rdfs:subPropertyOf rdf:resource="#accessEStoreThing"/>
   </owl:ObjectProperty>

   <!-- http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#Airline -->

   <owl:Class rdf:about="#Airline">
      <rdfs:subClassOf rdf:resource="#Travel"/>
      <owl:disjointWith rdf:resource="#Cruise"/>
      <owl:disjointWith rdf:resource="#Train"/>
   </owl:Class>
 <!-- http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#AppleProuduct -->

   <owl:Class rdf:about="#AppleProuduct">
      <owl:equivalentClass>
         <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
               <rdf:Description rdf:about="#IPhone"/>
               <rdf:Description rdf:about="#iPad"/>
               <rdf:Description rdf:about="#iPod"/>
            </owl:unionOf>
         </owl:Class>
      </owl:equivalentClass>
      <rdfs:subClassOf rdf:resource="#ElectronicDevice"/>
      <owl:disjointWith rdf:resource="#Audio_And_Video"/>
      <owl:disjointWith rdf:resource="#Cameras_And_Camcorder"/>
   </owl:Class>
<!-- http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#Zoppini_Bracelet_Jawelry -->

   <E_Stor_Thing rdf:about="#Zoppini_Bracelet_Jawelry">
      <rdf:type rdf:resource="#Bracelet_Jewelry"/>
      <rdf:type rdf:resource="#Jewelry"/>
      <rdf:type rdf:resource="&owl;Thing"/>
   </E_Stor_Thing>
      <!-- http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#Zyrtec_Allergy -->

   <owl:Thing rdf:about="#Zyrtec_Allergy">
      <rdf:type rdf:resource="#E_Stor_Thing"/>
   </owl:Thing>

   <!-- http://www.semanticweb.org/ontologies/2010/11/E-Store.owl#luxuryRoomPrice -->

   <Price rdf:about="#luxuryRoomPrice">
      <rdf:type rdf:resource="&owl;Thing"/>
   </Price>
</rdf:RDF>
</owl_class>
```

Fig. 1 A View of OWL-2 based Crisp Lightweight Ontology

In [12] seven steps of ontology development are discussed which covers almost every aspect of ontology modeling. These steps are:

1. Determine the domain and scope of the ontology
2. Consider reusing existing ontologies
3. Enumerate important terms in the ontology
4. Define the classes and the class hierarchy

5. Define the properties of classes—slots
6. Define the facets of the slots
7. Create instances

We developed heavyweight ontology for an electronic store (e-store) following the steps mentioned in [12]. The basic difference between lightweight and heavyweight ontology is that in heavyweight ontology, the ontology is enriched by axioms which produce a deep expressivity level. In the past, experiments based on lightweight ontology produced set of results with limited inferences. PAL (Protégé Axiom Language) is used to write axioms in Protégé (a well known ontology development tool by Stanford university) [14]. PAL is basically a constraint language that is used to enforce the semantic properties of knowledgebase encoded in Protégé [11]. These axioms are based on mathematical notation. They add constraints and link the information with other sets of information by applying certain criteria. The addition of axioms in lightweight ontology increases the expressivity level of ontology, and helps in machine interpretability. The PAL constraint rules for the electronic store ontology in which an *IPhone* price must be greater than *Nokia3200* can be coded as shown in Fig 2.

```
(defrange ?IPhone  :FRAME IPhone  )
(defrange ?Nokia3200 :FRAME Nokia3200  responsible_for)(forall
? IPhone   (forall ? Nokia3200
   (=> (and  (responsible_for ? IPhone   ? Nokia3200)
      (own-slot-not-null Price? IPhone  )
      (own-slot-not-null Price? Nokia3200))
   (> (Price? IPhone  ) (Price? Nokia3200)))))
```

Fig. 2 The PAL axioms Code File

IPhone is the name of the variable. Variable names start either with '?' or '%' which define local and global variables, respectively. *FRAME* is one type of variable. Other types of variables are *SET*, *SYMBOL*, *STRING*, *INTEGER*, and *NUMBER*. Fig. 3 shows the inferred model of the E-Store ontology. The inferred model is the graphical structure of interconnected domain elements. We used *Pellet Reasoner* Tool in Protégé to generate inferences; a screen view of *Palett Rasoner* can be seen in Fig. 6(a).

To add the fuzzy property into ontology, we mark the fuzzy terms in our ontology design before exporting heavyweight ontology into Fuzzy OWL tab of Protégé. In case of electronic mall we have several fuzzy concepts like price, quantity, size, delivery time etc. the fuzzy concept price has some fuzzy properties like *Price {Low Price, high price, Medium Price}* similarly we can define the fuzzy properties for the concept *size {Large, Medium, Small}*. Fuzzy OWL is basically a semiautomatic plug-in of protégé which assists the ontology author to add fuzzy logics in ontology. Fuzzy OWL-2 utilizes the annotation property of OWL-2 to entertain the fuzzy terms. Writing annotations manually is very tough and time consuming job. Also the ratio of error pruning in manual annotations writing is very high. Fuzzy OWL tab makes all the process speedy and error free. After installing plug-in we see a drop down menu on plug-in page .This helps us to create fuzzy data types, fuzzy concepts, fuzzy modifiers, fuzzy axioms etc. Fig. 4 is the screenshot of fuzzy OWL plug-in of Protégé 4.1. On the down right side of Fig. 4 we can see the annotations which we added to define the fuzziness of the terms [14].

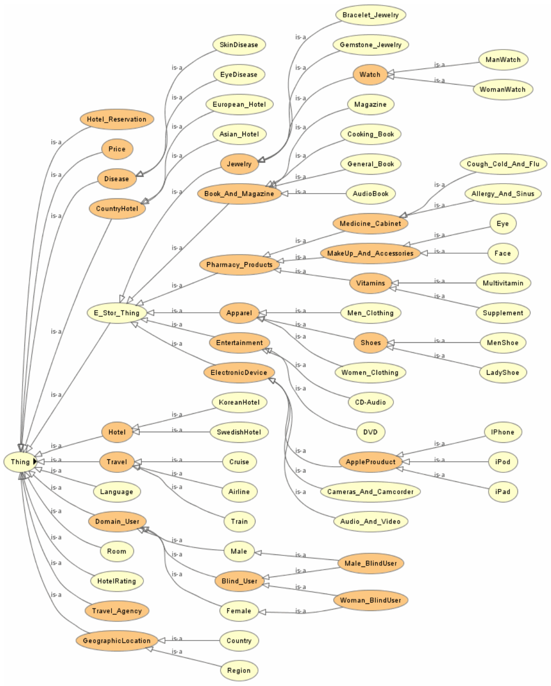

Fig. 3 E-Store heavyweight ontology inferred Model generated by Protégé OWL-VIZ extension

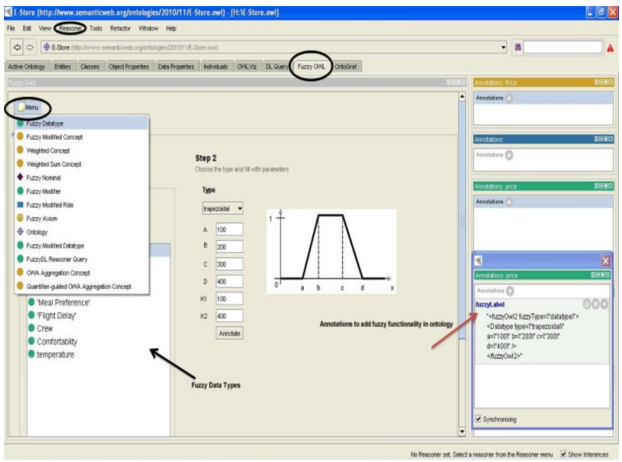

Fig. 4 A view of Fuzzy OWL plug-in which is used for fuzzy ontology modeling

3. Proposed Architecture for Imprecise Information Extraction for Decision Making

The proposed heavyweight fuzzy ontology based framework is a novel architecture which facilitates for extraction of highly precise information from imprecise heterogeneous web sources using a vocal command system for exact and timely decision making for the visually impaired user. The architecture is based on already developed technologies, so little effort is needed to implement it publicly. To enable visually impaired people globally to have the same quality of life as normally healthy citizens is a daunting task. The proposed architecture consists of seven functionality steps Fig. 5 shows the architecture.

3.1 At Step 1

At first step blind user has option to send his/her query through PDA, cell phone or PC. We assumed in our experiment that the visually impaired user had a smart phone with Pocket CMU SPHINX-2 [15] installed as an application on it. The user records his/her voice message in Pocket CMU SPHINX-2 speech to text conversion mode. The user's voice will automatically be converted into text, and sent to the step 2 for further processing. This phenomenon needs a real time continuous speech recognition system (RTCSRS) which should be effective, fast and must be lightweight so that mobile devices can easily use this with low power consumption. For this purpose we choose the Pocket CMU SPHINX-2 as it is already tested and analyzed by researchers several times in their experiments. The Pocket CMU-SPHINX2 is an open source large vocabulary continuous speech recognition system. There are many portable text to speech (TTS) and speech to text (STT) recognition systems available on internet under closed license but most of them are expensive and available without source code. The Pocket CMU SPHINX-2 is the first known open source continuous speech recognition system to date. Pocket CMU SPHINX-2 is lightweight, and specially developed for a mobile platform and low processing handheld devices. At the step 1, the visually impaired user records a voice message and the built-in Pocket CMU-SPHINX-2 encodes it into data packets [15]. These data packets are sent to the step 2, where the query builder module is waiting to handle the request.

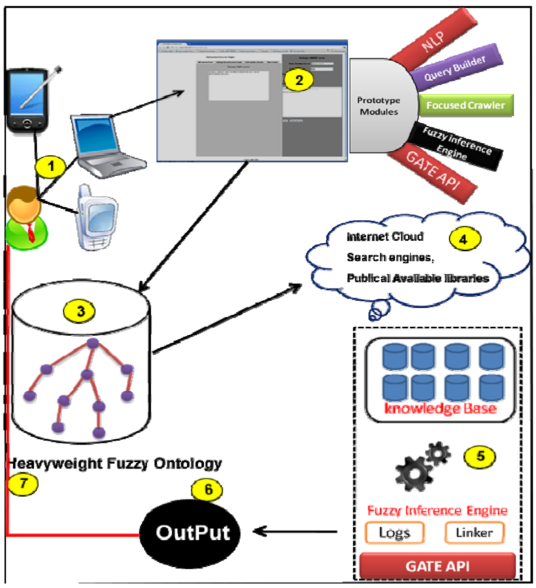

Fig. 5 Heavyweight Fuzzy Ontology based Prototype Architecture for Decision Making

3.2 At Step 2, 3 & 4

At this stage, the visually impaired user's text is received at prototype query handler module which is based on Ajax as shown in Fig. 6. Ajax is not a technology itself, but is instead a combination of JavaScript and XML. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. In Ajax, the server and client remain in connection with the help of an *XMLHttpRequest* function which is specially developed for seamless communication purposes [16]. The query handler module performs natural language processing, shallow parsing and converts the human language text into formal query. Due to Ajax it sends this query to focused web crawler automatically. The focused crawler is based on heavyweight fuzzy ontology which we have already developed in previous section. The major duty of crawler utility is to search information from internet and store it into knowledge base.

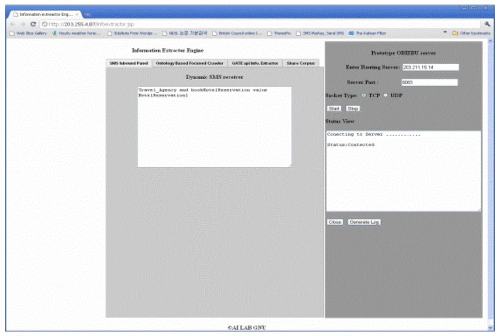

Fig. 6 Query Handler Module of Prototype

3.3 At Step 5, 6 &7

Relevant information extraction from the internet is always a difficult process in the information engineering domain. A web crawler is a program that browses the web data automatically. The difference between a simple web crawler and a focused web crawler is that a simpler crawler just searches the web and

Indexes the results which may be relevant or not. The focused crawler only indexes pages which belong to a particular domain, and which converge with certain predefined criteria [17]. The step 5 consists of knowledge base, fuzzy inference engine and GATE API. The knowledge base is populated by the heavyweight fuzzy ontology based crawler. Many web crawlers developed in the past had different criteria for searching based on page relevance, link relevance and so on. The server side dynamic web pages are based on databases, and these web page contents are populated in real time from the database and the populated values from the database changes according to the defined criteria. This knowledge base becomes the basis for fuzzy inference engine which is used to take decisions for the extraction of precise information.

GATE is generalized architecture for text engineering is used to parse the information. It arranges the information in the form of lexical order and performs name- entity tagging, part of speech tagging and converts the raw data into meaningful information. The prototype has ability to store each event into its dynamic logs file. These files can become helpful resource in troubleshooting of system as these keep records of all activities performed by user or system [18]. At the last stage results are forwarded to the user's handset and the built-in conversion module converts results into voice and user can hear the inquiry response in the form of voice. As this process takes not more than 180 sec (in ideal case, checked experimentally), so it can help the blind user to take decision quickly.

4. Experiments and Results

4.1 Research Design Evaluation and Validity

To ensure the effectiveness of our proposed research design in ontology based information systems. We have some known ways of evaluation, such as description logic queries, user level testing and domain expert manual evaluation. DL-Query and fuzzy DL-Query are powerful and user friendly tools used for searching a classified ontology. We tested and analyzed the performance of our modeled heavyweight ontology using Protégé 4.0. Manchester OWL syntax is used for writing description logic queries. *Reasoner* is a tool which can infer information on the basis of described rules. The Protégé Package normally has *FACT++* and *Pellet* as reasoner by default and the fuzzy OWL tab also has fuzzy ontology based reseasoners named as: *FuzzyDL* and *DeLorean* [14] .We used *Pellet* and FuzzyDL reasoners in our experiments. Pellet is one of the best tools which provide cutting edge reasoning for sound and complete information extraction. Pellet is available in the Protégé Package by default, or it can be added as a third party utility. Figure 7(a) shows the Pellet is working on electronic store ontology.

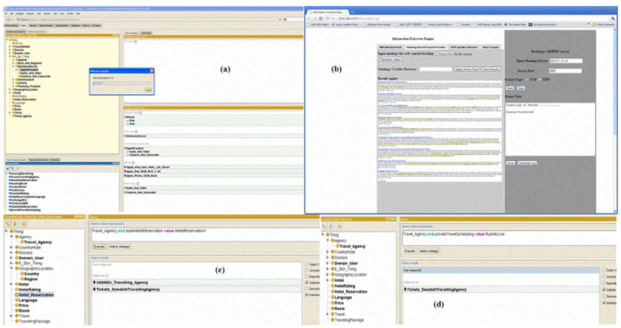

Fig. 7 Experiment Results on different Scenarios.

7(a) Pellet Reasoner tool in working mode. 7(b) Applet window of ontology based focused crawler. 7(c & d) Result windows of DL-queries

We designed some queries (with user and domain expert concern) to completely judge the overall performance of the system and found results according to requirements. Some of these are listed in table 1 and resulting screenshots can be viewed in 7(c) and 7(d).

4.2 System Precision Measurements

A series of experiments were designed to test and evaluate the efficiency and performance of the ontology, and the performance and information extraction capabilities of the prototype tool. Normally in every information extraction system there are three main performance measures: extraction rate, precision and recall. These parameters are considered while evaluating the performance of the system. Mathematically precision and recall can be expressed as:

$$RC = \frac{ce}{ce+te} \times 100\% \quad (3)$$

$$PR = \frac{ce}{ce+fe} \times 100\% \quad (4)$$

Where RC is the recall and PR is the precision 'ce' is the correct information that is extracted, 'te' and 'fe' represents the right and wrong information that are extracted by system respectively.

Table 1. A table depicting DL-Queries and explanation with possible Results

DL-Query Syntax	*Explanation*	*Possible Results*
Travel_Agencyan dprovideTravelScheduling value RyanAirLine	In this query the domain user wants information from a travel agency, whose schedule is selected according to the domain user's schedule.	Swedish Travel Agency
Travel_Agency and bookHotelReservation value HotelReservation 1.	This query displays travel agencies that can make hotel reservations.	XANADU and Swedish
Domain_User and accessJewelry value BraceletJewelry1	This query describes the user, who, for the purposes of our test, is the valid domain user and likes jewelry; specifically, bracelets.	Blind Female User

Our experimental environment had Intel(R) Core (TM1) 2 Quad CPU with 2.0 GB RAM, Window XP. We used Tomcat as web server and CDMA based external modem for sending and receiving user SMS. First, we designed the lightweight ontology and loaded it into our ontology crawler. This practice returned some statistical outcomes. Subsequently, we added the axioms and fuzzy logics into our ontology and refined it to get deep expressivity levels and loaded the ontology into the crawler and searched from the internet. We obtained different precision and recall results, shown in the tables below. Table 2 and 3 depict some statistics of ontology base information extraction tool.

Table 2 Precision and Recall Statistics of prototype using conventional E-Store ontology

Ontology Elements	*Resource Elements (ce)*	*True Elements (te)*	*False Elements (fe)*	*Precision (PR) (%)*	*Recall (RC) (%)*
Class (Concept)	413	389	24	94.5	51.5
Property	687	591	96	87.8	53.8
Individual	109	93	16	87.2	54.0

Table 3 Precision and Recall Statistics of prototype using E-Store heavyweight fuzzy Ontology

Ontology Elements	*Resource Elements (ce)*	*True Elements (te)*	*False Elements (fe)*	*Precisio n (PR) (%)*	*Recall (RC) (%)*
Class(Co ncept)	413	235	178	69.9	63.7
Property	687	496	191	77.3	58.1
Individual	109	78	31	77.9	52.3

We have noted that precision and recall ratios using a conventional ontology approach are: 69.9%, 77.3%, 77.9% and 63.7%, 58.1%, 52.3% respectively. In the case of the heavyweight fuzzy ontology, the precision rate increased from 69.9% to 94.5%, 77.3% to 87.8% and 77.9% to 87.2%. So according to our experiment more precise information can be achieved by using heavyweight fuzzy ontology. In figure 8, the bar chart shows a comparison between heavyweight fuzzy and lightweight ontologies on the basis of true and false elements. On the vertical side of graph, the metric 0 to 700 refers to elements retrieved by the ontology crawler's spider. *Classes, Properties* and *Individuals* on horizontal side refer to the measurement of the classification of results.

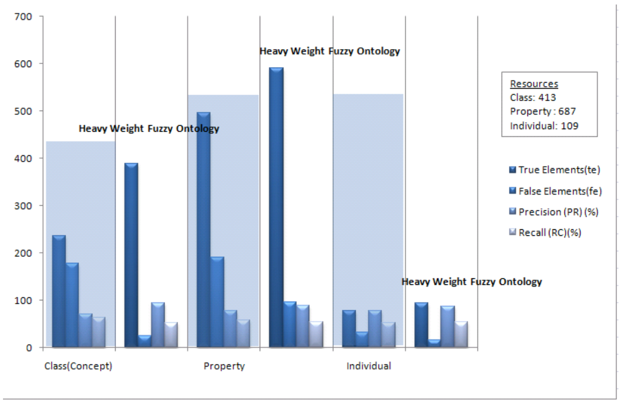

Fig.8 Graphical comparison between classical and Heavyweight fuzzy Ontology

5. Conclusion

In this research paper, we propose heavyweight fuzzy ontology architecture to extract exact information from imprecise resources for timely decision making. The automatic execution of this novel architecture makes the complex process of vague information extraction faster and more adaptable than existing solutions. The interactivity of the prototype enables the user to retrieve and filter the desired information while roaming using cell phone and PDA. The Classical ontology technique can not deal with blur information due to its limitation in concepts expressivity and intelligence. The proposed technique ideally copes to extract information in several kinds of data. We presented the results and performance measures of our tool and ontology. The results were significantly better than our previous experiments. The enrichment of axioms and fuzzy logic produces a quality of expressivity with intelligence, and makes the information more understandable for computers. As a result, the precision of extracted information increases which further helps to take right decisions. Currently, we are developing new capabilities for automatic axioms rules generation with fuzzy type-2 logic, and improving the portability of this system. More research is required, but we are confident that this will make the system more efficient overall.

References

[1] Chang, C., Kayed, M., Girgis, M.R., and Khaled F., "A Survey of Web Information Extraction Systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1411-1428, October 2006.

[2] Crescenzi, V., Mecca, G., "Automatic information extraction from large websites," *Journal of the ACM (JACM),* vol. 51, pp. 731-779, September 2004.

[3] Kushmerick, N., "*Wrapper induction: Efficiency and expressiveness*," Department of Computer Science, University College Dublin, Dublin 4, Ireland 2003.

[4] Brown, M.K., Glinski, S.C., and Schmult, B.C., *Web Page Analysis for Voice Browsing, Interaction Design: Beyond human-computer interaction*, John Wiley & Sons, 2007.

[5] Zajicek, M., Powell, C., and Reeves, C., "A Web navigation tool for the blind," *Proceedings of the third international ACM conference on Assistive technologies*, pp. 204-206, April 15-17, 1998, Marina del Rey, California, United States.

[6] Shadbolt, N., Hall, W., and Berners-Lee, T., "The Semantic Web Revisited," *Intelligent Systems*, IEEE, vol. 21, no. 3, pp. 96-101, 2006.

[7] Gerber, A.J., Barnard, A., and van der Merwe, A.J., "Towards a semantic web layered architecture," *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*, pp. 353-362, February 13-15, 2007, Innsbruck, Austria.

[8] Jun, Z., Yang, Li., Qinglian Wang., and Miao L., "Knowledge Sharing for Supply Chain Management Based on Fuzzy Ontology on the Semantic Web", *Information Processing (ISIP), 2008 International Symposiums on*, pp. 429-433, 23-25 May 2008.

[9] Sun, Yi., Zhang D., and Cheng., Li., "Fuzzy ontology constructing and its application in traditional Chinese medicine", *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 2, no. 2, pp. 592-595, 29-31 Oct. 2010.
[10] Park, H., Lee, S., "Similarity Analysis Between Fuzzy Set and Crisp Set," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7, no. 4, pp. 295-300, 2007.
[11] Kim., K., Kim, C., Moon, J., "A Study on Performance Assessment Methods by Using Fuzzy Logic", *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 3, no. 2, pp. 138-145, 2003.
[12] Natalya F.N., and Deborah L. McGuinness., "Ontology Development 101: A Guide to Creating Your First Ontology," http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html, 2001.
[13] W. Grosso, "The Protégé Axiom Language: Overall Design Considerations," http://protege.stanford.edu/plug-ins/paltabs /OverallDesignConsiderations.zip, 11.10.2009.
[14] Fernando, B., Straccia., U, "Fuzzy Ontology Representation using OWL 2," *International Journal of Approximate Reasoning Online Printed*, May 2011.
[15] Daines, H., Kumar, M., and Chan, A., "Pocket -Sphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices," *Proceeding of ICASSP*, pp. 41-46. Slovakia, September 2005.
[16] Meng, H., Li, Y.C., Fung, T.Y., and Low, K.F., "Bilingual Chinese/English voice browsing based on a VoiceXML platform," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, no. 2, pp. 769-772, 17-21 May 2004.
[17] John, H. Gennari., Mark, A. Musen., Ray, W. Fergerson., William, E. Grosso., Monica, C., Henrik E. , Natalya, F. Noy , Samson., and W. Tu., "The Evolution of Protege: An Environment for Knowledge-Based Systems Development," *Int. J. Human-Computer Studies*, vol. 58, no. 1, pp. 89-123, 2003.
[18] Gatial, E., Balogh, Z., Ciglan, M., and Hluchy, L., "Focused Web Crawling Mechanism based on Page Relevanc," *In Proceedings of ITAT 2005 Information Technologies Applications and Theory*, pp. 41-46, September 2005.

Ahmad C. Bukhari received the B.S. degree in Mathematics from University of Punjab, Lahore in 2005 and M.Sc. Degree in information Technology in 2008 from PUCIT, Lahore Pakistan. Currently he is perusing M.S. degree from Gyeongsang National University Korea. His area of research includes intelligent system, semantic web, soft computing, autonomous underwater vehicles and intelligent ships

E-mail: ahmad@gnu.ac.kr

Yong-Gi Kim received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea in 1978, the M.S. degree in computer science from University of Montana, U.S.A. in 1987, and the Ph.D. degrees in computer and information sciences from Florida State University, U.S.A. in 1991. He was a Visiting Scholar in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, U.S.A., from 2008 to 2009. He is currently a Professor in the Department of Computer Science, Gyeongsang National University, Korea. His current research interests include soft computing, intelligent systems and autonomous underwater vehicles.

Corresponding author: E-mail: ygkim@gnu.ac.kr