ISSN: 2321-8169
782 – 788

Ontology Based Contextual Intent Recognition for Speech Gesture Utterances

Ms.R.Saktheeswari
PG Scholar, IT Department
Sri Venkateswara College of Engineering.
Sriperumbudur,Tamil Nadu,India
sakthi@svce.ac.in

Ms.N.Devi
Assistant Professor,IT Department
Sri Venkateswara College Of Engineering
Sriperumbudur,Tamil Nadu,India
nds@svce.ac.in

Abstract— Recent developments in speech, network and embedded-computer technologies indicate that human-computer interfaces that use speech as one or the main mode of interaction will become increasingly prevalent. Such interfaces must move beyond simple voice commands to support a dialogue-based interface To support human-computer dialogue effectively, architectures must support active language understanding. It is generally accepted that using context in conjunction with a human input, such as spoken speech, enhances a machine's understanding of the user's intent as a means to pinpoint an adequate reaction. The contribution of this work is to provide empirical evidence of the importance of conversational context in speech-based human–computer interaction. A framework for context-sensitive computing approach is presented, to address how to extract contexts from speech, how to process contextual entities by developing an ontology-based context model and how to utilize this approach for real time decision making to optimize the performance indicators .

Keywords:HCI,intent recognition,conrexrual autolearning,speech gesture.

*****

I. Introduction

Natural language processing is a difficult and increasingly important topic to study. With computers becoming more and more "knowledgeable" and assisting in some everyday tasks, the need for an accurate natural language processing system becomes more apparent and desirable. Recognizing the intent of an utterance is a difficult task. To understand the intended meaning without additional hints, it may be necessary to recognize many meanings of the utterance and then choose the most appropriate one for the situation. The choice may be made using heuristics. It is not always an easy task for a human, let along the computer, to recognize the meaning of an utterance. One difficulty is because the utterance can have both a literal and a non-literal meaning.

Human speech provides a natural and intuitive interface for communication [2]. Recognizing the intentions of others is an important part of human recognition. Although the understanding of spontaneous spoken language is an open issue in natural language processing and artificial intelligence, in practice, the concept of understanding is situation-dependent. I n addition, more accurate identification is needed for multiple services in order to distinguish different speech acts of utterances with different word order. The purpose of this paper is to present an ontology-based approach of context-sensitive computing for the optimization.

Previous works on intent recognition [8] are based on either incorporating confidence scoring or discourse information. Initial works on intent recognition uses HMM models in which intent recognition is about prediction. Later on work begins to address these issues, and introduces the idea of using a digraph based language model to provide contextual knowledge. In another case a statistical model to predict speakers' intentions by using multi-level features was proposed. Using the multi-level features like morpheme-level features, discourse level features, and domain knowledge-level features, the model predicts speakers' intentions that may be implicated in next utterances. Yet another method uses RE mechanism as a dynamic and personalized framework for human-computer collaboration. In this framework, a human dynamically interacts with a computer partner by communicating through the haptic channel to trade control levels on the task.

Ontology are a widely accepted tool for the modeling of context information. Ontological analysis clarifies the structure of knowledge. Given a domain, its ontology forms the heart of any system of knowledge representation for that domain [10]. Without ontology, or the conceptualizations that underlie knowledge, there cannot be a vocabulary for representing knowledge. Although the understanding of spontaneous spoken language is an open issue in natural language processing and artificial intelligence, in practice, the concept of understanding is situation-dependent. In addition, more

ISSN: 2321-8169
782 – 788

accurate identification is needed for multiple services in order to distinguish different speech acts of utterances with different word order.

An ontology-oriented approach can be used to build context models to support context awareness in pervasive computing environments. Ontology and contexts are complementary disciplines for modeling views. In the area of information integration, ontology may be viewed as the outcome of a manual effort of modeling a domain, while contexts are system generated models. The paper is organized as follows chapter 2 gives a brief introduction about the architecture of our system. In chapter 3 the various elements of our architecture was explained, followed by details about implementation in chapter 4.

.

II. PROPOSED SYSTEM ARCHITECTURE

The use of context is important in interactive applications. It is particularly important for applications where the user's context is changing rapidly[7]. Computing devices and applications are now used beyond the desktop, in diverse environments, and this trend toward ubiquitous computing is accelerating. One challenge that remains in this emerging research field is the ability to enhance the behavior of any application by informing it of the context of its use. By context, we refer to any information that characterizes a situation related to the interaction between humans, applications and the surrounding environment. Context-aware applications promise richer and easier interaction, but the current state of research in this field is still far removed from that vision. Difficulties arise in the design, development and evolution of context-aware applications. Designers lack conceptual tools and methods to account for context awareness. As a result, the choice of context information used in applications is very often driven by the context acquisition mechanisms available. RDF, Ontology and SPARQL form the three core component of context aware applications.

We incorporate modular ontology, where each ontology module represents a domain and can be dynamically loaded at runtime to meet the current needs of the user. In order to provide a personalized answer, tailored to the specific user, the concepts and attributes in these ontology modules are annotated with scores representing the preferences and interests of the user. This allows us to learn the specificities of a user, and give responses that fit the user's profile. Also, this provides us with the building blocks for constructing ad-hoc communities of similar users where information can be shared and recommendations can be made.

A. RDF

RDF is the first language developed especially for the Semantic Web. It is recommended by W3C for writing machine process able annotations. RDF defines resources using XML. RDF is also called triple because it has three parts subject, object and predicate. Subject and object are names for resources and predicate is the relationship that connects these two things. All Information can be represented in the form of triples. RDF represents relationship between any two data elements, allowing for a very simple model .

B. ONTOLOGY

Ontology describes the conceptualization, the structure of the domain, which includes the domain model with possible restrictions. More detailed ontology can be created with Web Ontology Language (OWL). It is syntactically embedded into RDF, so like RDFS, it provides additional standardized vocabulary. For querying RDF data as well as RDFS and OWL ontology with knowledge bases, a Simple Protocol and RDF Query Language (SPARQL) is available [3]. SPARQL is SQL-like language, but uses RDF triples and resources for both matching part of the query and for returning results .specification of a shared conceptualization [3]. Ontology is specific to a domain, and it represent *an area of* knowledge Hence users and domain experts should agree on the knowledge being represented by ontology so that it can be shared and reused.

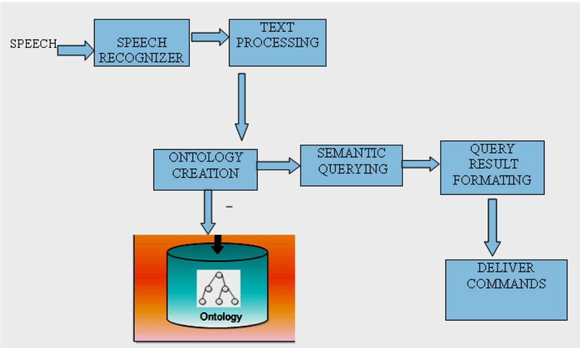

Figure 1. Contextual Auto learning architecture.

C. SPARQL

SPARQL was standardized by W3C. SPARQL is a query language that is used to query RDF data. It can also be used to query remote RDF server. Like RDF, basic building block of SPARQL query is the triple pattern. A triple pattern is like a triple, but it can have variables in place any of the three positions: subject, predicate or object.

III. The Ontology Based Context Modeling

Ontology are a widely accepted tool for the modeling of context information. One of the key factors for accurate and effective information access is the user context. The critical elements that make up a user's information context include the semantic knowledge about the domain being investigated, the
Short-term information need as might be expressed in a query, and the user profiles that reveal long-term interests. We propose a framework for contextualized information access that seamlessly combines these elements in order to effectively locate and provide the most appropriate result for users' information needs. In particular, we focus on integrating a user's query with semantic knowledge from an existing concept hierarchy to assist the user in information retrieval. In our framework, the user's "context" is captured via nodes in a concept lattice induced from the original ontology and is updated incrementally based on user's interactions with the concepts in the ontology.

A. Speech Recognition

Speech recognition (SR) in terms of machinery is the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words[6]. To build a general purpose speech recognizer, a huge amount of data would have been needed which was not feasible in this short period of time, so before proceeding with the continuous speech recognizer, a domain has been chosen . The following are the steps involved in building a speech recognizer using Sphinx 4

DATA PREPARATION

For the our application we have collected the different possible ways of expressing the commands for editing from different user and the same has been stored in a text file.

BUILDING SPEECH CORPUS AND GRAMMAR

After selecting text for the recognizer, recording of this chosen data is required. For this work, the recording has been taken place using different speakers. A grammar or a language model is used to specify all valid sentences in your speech recognition application.

BUILDING DICITONARY FILE

A language dictionary file where each word maps to a sequence of sound units to derive the sequence of sound units associated with each signal. The file should contain unique entry and sorted accordingly. The file is saved with .dic extension.

BUILDING LANGUAGE MODEL

The language model, LM file, describes

the likelihood, probability taken when a sequence or collection of words is seen. To build this file, CMU lmtoolkit is used. Lmtool is a web based tool that allows

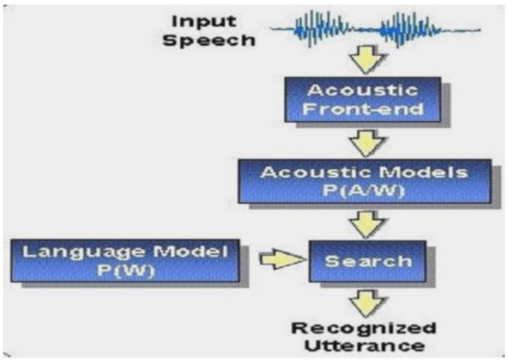

Figure.2 Basic model of speech recognition

users to quickly compile two text-based components needed for using an ASR decoder. Upload this file, click the compile button. This will give a set of lexical (pronunciation dictionary) and language modeling files. Here the only file used is LM file as Pronunciation dictionary should be built as stated above.

B. Text Processing

Text categorization is the task of assigning predefined categories to free-text documents. It can provide conceptual views of document collections and has important applications in the real world [5]. The different phases in text categorization includes the following

During the first pass, the tagger assigns a part of speech tag to each word in the corpus. During the second pass, all nouns and verbs are looked up in Word Net and a global list of all synonyms and hypernym synsets is assembled. Infrequently occurring synsets are discarded, and those that remain form the feature set. A synset is defined as infrequent if its frequency of occurrence over the entire corpus is less than 0.05N, where N is the number of documents in the corpus. During the third pass, the density of each synset defined as the number of occurrences of a synset in the Word Net output divided by the number of words in the document is computed for each text resulting in a set of numerical feature vectors.

MAPPING TERMS INTO CONCEPTS

The process of mapping terms into concepts is illustrated with an example shown in Figure 2. For simplicity, suppose there is a text consisting in only 10 words: government (2), politics (1), economy (1), natural philosophy (2), life science (1), math (1), political economy (1), and science (1), where the number indicated is the number of occurrences.

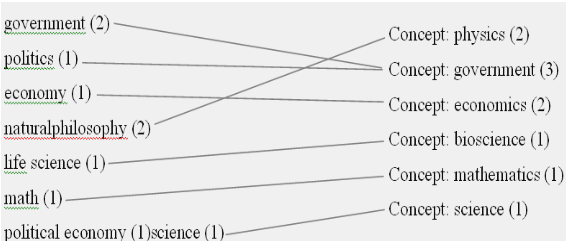

Figure3. Example of mapping terms into concepts.

The words are then mapped into their corresponding concepts in the ontology. In the example, the two words government (2) and politics (1) are mapped in the concept government.

C. ONTOLOGY BUILDING

In order to build ontology, two different processes are required which make use of the annotations provided by the information extraction process. First, a preparatory module starts the ontology building and then an ontology builder completes the process by writing an OWL-file.

We use OWL-DL, a semantic markup language with expressive power and inference capability, to support semantic interoperability in context-embedded environment among various entities.
OWL uses an object-oriented approach to describe the structure of a domain in terms of classes and properties. Classes represent important objects or entities and individuals are instances of classes. The built-in OWL Property owl: subClassOf allows us to structure the hierarchical relationship between super-class and sub-class entities. Further, each class is associated with its attributes through Data type property or other entities through Object property. Data type properties link an individual to an XML-schema data type value while Object properties link an individual to an individual.

Ontology-based reasoning infers implicit contexts from explicit contexts based on class relationships and property characteristics. Standard reasoning rules that support OWL-DL entailed semantics can be defined for relationships like subclass Of, subPropertyOf, disjoint With, inverse Of, Transitive-Property, and Functional-Property.

ONTOLOGY PREPARATION

Based on the annotations from the information extraction component, the ontology preparation module identifies the structures, such as super- and subclasses, individuals, labels, properties and relations, which will subsequently be used to build the ontology. The preparation module also detects properties of individuals or whole classes [9]. Relations to other objects, such as “part of”, are identified by the sentence’s verb and annotated Relation with Meta information about the respective object. Relations between individuals are referred to as object properties, while properties of string-, integer- or Boolean values are called data type properties. Once the preparation is finished, the ontology can be built. This is done by the subsequent module called ontology builder.

ONTOLOGY BUILDER

The ontology builder requires the previously calculated information from the ontology preparation module [3]. It uses the Protégé’s OWL application programming interface to create an OWL-file. The information from class annotations is used to generate the ontology’s taxonomy. Super classes in the taxonomy can bear properties which are inherited to all individuals in all subclasses below that class,. Fixed values like data types can be defined for these properties, too. Sometimes a new class may be identified by the ontology builder which belongs just between two already existing classes.

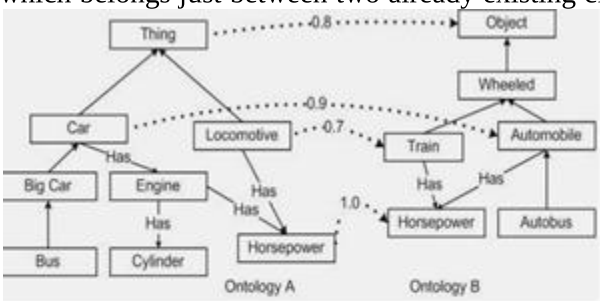

Figure 4 Concept hierarchy in Context Ontology

Therefore, the ontology builder is able to paste a new class in the so far built taxonomy. Individuals are attached to their respective classes and each individual inherits all properties from all its super classes.

ISSN: 2321-8169
782 – 788

Properties are also preconfigured with values at class level. If there are individuals with the same name but different meanings, a unique ID is assigned to each individual. This enables homonymy and polysemy as well as ontology naming restrictions.

With the described text mining component, a domain-specific ontology can be build .

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
 xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#
 xmlns:owl=http://www.w3.org/2002/07/owl#
xmlns="http://www.xfront.com/owl/ontologies/camera/#
xmlns:camera="http://www.xfront.com/owl/ontologies/camera/#"
xml:base="http://www.xfront.com/owl/ontologies/camera/">
 <owl:Ontology rdf:about="">
 <rdfs:comment> Camera OWL Ontology
 Author: Roger L. Costello
  </rdfs:comment>
 </owl:Ontology>
 <owl:Class rdf:ID="Money">
 <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
 <owl:DatatypeProperty rdf:ID="currency">
 <rdfs:domain rdf:resource="#Money"/>
 <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
 <owl:Class rdf:ID="Range">
 <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
 <owl:DatatypeProperty rdf:ID="min">
 <rdfs:domain rdf:resource="#Range"/>
 <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="max">
 <rdfs:domain rdf:resource="#Range"/>
 <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
 </owl:DatatypeProperty>
 <owl:DatatypeProperty rdf:ID="units">
 <rdfs:domain rdf:resource="#Range"/>
<rdfs:range
rdf:resource="http://www.w3.or/XMLSchema#string"/>
 </owl:DatatypeProperty> <owl:Class
rdf:ID="Window">
<rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
 </owl:Class>
```

D. SEMANTIC SEARCHING

Our system takes as input a formal ontology-based query, for which SPARQL and RDQL are currently supported. The predominant query language for RDF graphs is SPARQL. SPARQL is an SQLlikelanguage, and a recommendation of the W3C as of January 15, 2008.

SPARQL (pronounced "sparkle", a recursive acronym for SPARQL Protocol and RDF Query Language) is an RDF query language, that is, a query language for databases, able to retrieve and manipulate data stored in Resource Description Framework format.SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

An example of a SPARQL query to show the type of crop which grows in cool climate, using a fictional ontology.

```
SELECT ?ID ?Title ?Funding_Body ?PI ?Administartive_Authority ?Duriation ?Cost_in_Lakhs ?StartDate ?Status where {?ID rdf:type :Project; :hasTitle ?Title; :hasPI ?p; :hasAA ?a; :sponsorBy ?f. ?f organization:hasName ?Funding_Body. ?p people:hasName ?PI. ?a people:hasName ?Administartive_Authority. ?ID :hasDuration ?Duriation; :hasCost ?Cost_in_Lakhs; :hasStartDate ?StartDate; :hasStatus ?Status. }
```

SPARQL RESULT

ID	Title	Funding_Body	PI	Administartive_Authority	Duriation	Cost_in_Lakhs	StartDate	Status
P0002	Rooting System	DRDO	Prof. M. R. Naidu	Prof. K. Lakshmi	3 Years	4.85600014E1	2011-05-12T00:00:00	On going
P0004	Knowledge Representation System	DRDO	Prof. Mohanlal	Dr. Gurdeep	2 Years	4.05600014E1	2006-05-26T00:00:00	Completed
P0005	Resource Management System	ISRO	Dr. A. Usha Rani	Dr. M. Jawahar	3 Years	7.52600021E1	2007-01-24T00:00:00	Completed
P0001	Managing Cloud in the Internet	MOES	Dr. L. Kanhaiya Lal	Dr. Ajaipal	2 Years	1.05E1	2009-03-24T00:00:00	Completed
P0003	Knowledge Managment System	MOES	Prof. Dimpu Rani	Prof. M K Naidu	3 Years	3.03999996E1	2011-06-25T00:00:00	On going

Figure 5. SPARQL query result.

The SPARQL query language for RDF has several query result forms such as CSV, TSV and XML etc. These formats are not clear to the user to realize and analyze query results. There is a need to represent SPARQL query result in an attractive format so that user can easily understand the SPARQL query results. The SPARQL query result formats require additional conversions or tool support to represent query results in user readable format[4]. The method used should enable the user to build HTML document dynamically for variable binding SPARQL query results and browse the constructed HTML document automatically to view the query result. The query is executed on semantic data stored using the oracle Jena adapter. The HTML file construction section lists the variables involved in the query and extracts the variable binding values. A HTML Document is constructed with the variables and query results[10] . Finally the constructed HTML document is displayed to view the report using any web browser like internet explorer, etc.

SPARQL QUERY EXECUTION

Executing SPARQL query using Jena adapter has the following steps.

Create model for the ontology store

Take SPARQL query string from the interface

Create query using Query Factory.

Pass query and model to the QueryExecutionFactory and execute the query

List out the variables involved in the query

Extract variable binding values (it is an iterative process)

Constructing HTML document with the listed variables and binding values.

IV. IMPLEMENTATION

The proposed architecture is implemented using Java Net Beans IDE and Jena API. A simple user interface is designed for writing SPARQL query using Java Net Beans IDE. Implementation of the proposed method has two phases. First phase consist query execution process 7using the Oracle Jena adapter and the second phase describes generation of HTML document for SPARQL query results. To evaluate the proposed method the ontology was created using Protégé which provides facilities for creating classes sub-classes relationship. Protégé is a free, open source ontology editor and a knowledge acquisition system. Like Eclipse, Protégé is a framework for which various other projects suggest plug-in. This application is written in Java and heavily uses Swing to create the rather complex user interface.

Jena is an open source Semantic Web framework for Java. It provides an API to extract data from and write to RDF graphs. The graphs are represented as an abstract "model". A model can be sourced with data from files, databases, URLs or a combination of these. A Model can also be queried through SPARQL and updated through SPARQL. We use Word Net for text processing; Word Net is a lexical database for the English language[1]. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

V. EXPERIMENTS AND RESULTS

In order to carry out an evaluation of the system, we have asked colleagues at the institute AIFB to provide queries in the way they would interact with a system capable of processing keyword based queries, along with the natural language description of the query. These queries were incorporated only as an evaluation set and not used for the development or tuning of the approach. A query generated by our approach is regarded as correct if it retrieved the same answers as the hand crafted query. we evaluate the approach in terms of precision, recall and F-Measure. Precision P is defined as the number of correctly translated keyword queries divided by the number of cases for which the system was able to construct a query. Recall R is defined as the number of correctly translated keyword queries divided by all the keyword queries of the evaluation set. The F1 = (2————————*P*————————R)/(P+R) measure is then the harmonic mean between precision and recall.

VI. CONCULSION

Understanding intentions in context is an essential human activity, and with high likelihood will be just as essential in any System that must function in social domains. In our work we investigated the problem of modeling the intention of user utterances produce during interactive applications .We presented a method to model the intent of a user in human-computer interaction by modeling the intention using contextual information. Incorporating Word Net knowledge into text representation that can lead to significant reductions in error rates on certain types of text classification tasks Using Ontology for modeling the intention enables the system to respond naturally for different interactions. In the experiments, the proposed model showed better performances than the previous model. Based on the experiments, we found that the proposed ontology model is very effective in speaker's intention recognition.

VII. FUTURE ENHANCEMENT

The future work of our group will consider implementing new mechanisms for linking the generic multimodal ontology and affective interfaces with recent research in Semantic Web and HCI .As user contexts are

ISSN: 2321-8169
782 – 788

an important part of a dialog system, we are planning to learn new user contexts, which can be represented in the ontology by the DOLCE module Descriptions and Situations. Furthermore our goal is, to integrate the ontology learning framework into the open-domain spoken dialog system.

REFERENCES

[1] Zakaria Elberrichi1, Abdelattif Rahmoun2, and Mohamed Amine Bentaalah1,King Faisal University, Saudi Arabia,"Using WordNet for Text Categorization",The International Arab Journal of Information Technology, Vol. 5, No. 1, January 2008.

[2] Jingjing Liu, Stephanie Seneff, and Victor Zue, IEEE Member."Harvesting and Summarizing User-Generated Content for Advanced Speech-Based HCI",IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 6, NO. 8, DECEMBER 2012

[3] B. Chandrasekaran and John R. Josephson, Ohio State University,V. Richard Benjamins, University of Amsterdam,"What Are Ontologies, and Why Do We Need Them?" ,1999 IEEE IEEE INTELLIGENT SYSTEMS.

[4] Dr Sunitha Abburu, G.Suresh Babu ," Format SPARQL Query Results into HTML Report" , International Journal of Advanced Computer Science and Applications, Vol. 4, No. 6, 2013 .

[5] Mohamed Yehia Dahab , Hesham A. Hassan , Ahmed Rafea ,"TextOntoEx: Automatic ontology construction from natural English text",ScienceDirect, Expert Systems with Applications 2008,.

[6] M.A.Anusuya,S.K.Katti,"Speech Recognition by Machine: A Review",International Journal of Computer Science and Information Security, Vol. 6, No. 3, 2009.

[7] Richard Kelley, Alireza Tavakkoli, Christopher King, Amol Ambardekar, Monica Nicolescu, and Mircea Nicolescu ,"Context-Based Bayesian Intent Recognition",IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, VOL. 4, NO. 3, SEPTEMBER 2012.

[8] Nancy G. Leveson, Member, IEEE ,"Intent Specifications: An Approach to Building Human-Centered Specifications",IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 26, NO. 1, JANUARY 2000.

[9] A. Brasoveanu, A. Manolescu, M.N. Spînu,"Generic Multimodal Ontologies for Human-Agent Interaction",Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. V (2010), No. 5, pp. 625-633.

[10] Lee Begeja, Harris Drucker, David Gibbon, Senior Member, IEEE, Patrick Haffner, Zhu Liu, Senior Member, IEEE, Bernard Renger, and Behzad hahraray, Senior Member, IEEE ." Semantic Data Mining of Short Utterances", IEEE Transactions on Speech and Audio Processing, VOL. 13, NO. 5, SEPTEMBER 2005 .

[11] http://protege.stanford.edu/doc/users.html