OnSSA: An Ontology-based Semantic Search Agent in Medicine

Jung-Jin Yang, Chae-Myung Lim, Tae-Young Park, and DongHoon Lee
School of Computer Science and Information Engineering
The Catholic University of Korea
YuckGok 2-Dong San 43-1WonMi-Gu BuCheon-Si KyungGi-Do, Korea
+82-32-340-3377
{jungjin|cmlim|typark|dhlee}@catholic.ac.kr

ABSTRACT
The volume of literature in the medical domain is expanded exponentially and generating a proper query for finding related information puts a cognitive burden on users, due to a lot of professional keywords in the domain. We describe an ontology-based information retrieval agent system in medicine through bio-related literature database MEDLINE, in particular. The task of the system here is to proactively help user to reformulate queries in order to get useful and relevant information by utilizing both existing medical ontologies and its own ontology. This work presents Semantic Web and Agent integration.

Keywords
Information Retrieval, Semantic Web and Agent Integration, Semantic Search

INTRODUCTION

Information retrieval is an everyday activity. It can be a time consuming and cognitively demanding task due to the information overload and its complexity through the advent of the Internet. The difficulty of retrieving relevant information increases further in the professional domains such as medicine. Formulating adequate query itself imposes a severely heavy cognitive burden on users, due to the utilization of a lot of professional keywords. While the need to help users reduce their work and to improve search results has emerged, methods for systematic retrieval and adequate exchange of relevant information are still in their infancy.

The recent report of completing human genome project indicates a potential shift of preventing and treating paradigm of disease. It is known to provide the foundation of leaping current medicine from experience-based and information-based ones toward prediction-based medicine. However, achieving the beneficiary goal still needs a way to retrieve and analyze the exponentially expanding bio-related information for finding correlation among information with various and sensitive factors. Moreover, research on bio-related fields naturally applies knowledge discovered to the current problem and make inferences to extract new information. It is necessary for researchers to communicate and exchange knowledge in order to extract further knowledge based on shared concepts. The shared concepts and data containing information need to be defined and used in a coherent way. Ontologies that specify terms and relationships among terms [1][2][3] facilitate sharing of data and knowledge among computational biologists. Also expanded volume of bio-related literature in MEDLINE needs a systematic search strategy and reviews, for the literature contains inter-related information and newly discovered knowledge in it. However, it is published in an electronic form that is not accessible by machines and that makes it hard to gain correlated information from it.

We employ the Semantic Web technique for augmenting targeted data with markup that describes some meaning of the data and encodes it in a form that is suitable for machine understanding. The Semantic Web community addresses these issues by defining standard mark-up languages like RDF, RDF Schema, DAML+OIL, and OWL[4]. These languages provide features to represent both shared concepts in an ontology and data in a form to be processed.

We have built an ontology-based information retrieval agent system in medicine through bio-related literature database MEDLINE, in particular. The goal of this research is both to improve the quality of information retrieval and to reduce user's cognitive load during information search. This paper describes our preliminary design and implementation of such a system called OnSSA (an Ontology-based Semantic Search Agent) that automates systematic retrieval of literature in medicine by utilizing ontology-based query models.

Most users make a query on general-purpose search engine without having professional knowledge about what they are looking for. It is rather an opportunistic search instead. In order to gain effective information retrieval, users check on results of previous query returned from search engine and reformulate a query either a bit too

general or specific. For example, suppose a general user or an expert puts a query with 'deafness' to PubMed interface, a biomedical search engine. The system returns 33033 numbers of documents as results retrieved from MEDLINE database. Unlike general-purpose search engines, professional engines of retrieving professional literature references are often hard to formulate queries without having proper knowledge of the domain.

To resolve this problem, our agent system utilize agent ontology to generate a *relevant* query of a keyword given by a user utilizing existing medical ontologies as well. MEDLINE is an on-line bibliographic database created by the U.S. National published since 1965. Multiple interfaces are available for searching MEDLINE; each differs in appearance and internal logic but seeks the same target content. While these interfaces are valuable to some degree, unfortunately, the few published strategies for identifying these articles involve MEDLINE interfaces not widely available outside of academic medicine [5][6][7]. Therefore, we retrieve literature through PubMed that is a MEDLINE interface freely available via the Internet from the NLM.

FRAMEWORK

We start this section by describing the overall process in OnSSA and then describe in detail how the agents help the system build the adapted query.

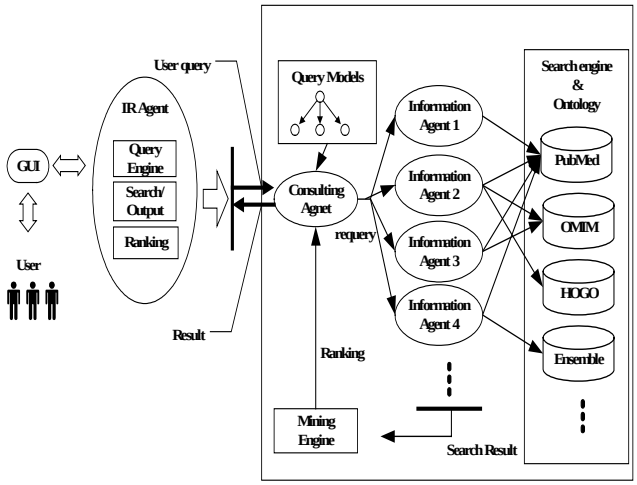

Fig. 1 System Overview

OnSSA consists of four modules as shown in Fig. 1. The *input module* proactively constructs the queries on behalf of the user as it utilizes existing medical ontologies and query models of the system. For example, if a user queries with a disease name, then the interface agent lets a query be a disease name with relevant gene names. It is then confirmed with a user to make sure the query regenerated is what the user wants to search. A consulting agent is responsible for interacting with the user. It refers to the query model to generate a relevant query, gets a confirmation on the query reformulated from the user, finds an agent list for matching the query in hand and assigns the query to distributed information agents, and recommends selected information to the user.

The ontology built on these Semantic Web standard languages could include terminologies of this field – class taxonomy in RDF Schema, properties defined for the classes can model a schema of an agent's mental state, the state itself is represented by an RDF graph, it comprises facts defined a priori as well as knowledge acquired through the perception system. The information agents take different semantic search strategies for information retrieval.

User interface accepts the user's query and transfers it to the IRagent system in Fig. 1. The IRagent system contains three modules: *Query engine* module to reformulate a query given, *Search/output* module to orchestrate systematic search using distributed information agents, and *Ranking* module to take over and rank the results retrieved by the information agents. User's query is sent to the Query managing agent. It is transferred to the Prolog-like clause in the first order logic (FOL) form. Query managing agent uses ontology network and generate relevant queries through inferences. Reformulated queries are confirmed through user's feedback and distributed to information agents. The kinds of queries given to information agents can be searches for disease, disease-gene, disease-gene-protein relations and more. Information agents use different biomedical ontology depending on either different queries or agents' search strategies.

Task or context-specific analysis of biological data requires exploiting the relations between terms used to specify the data, to extract the relevant information and to integrate the results in a coherent form. Biomedical information is rather well-defined in terms of classification and taxonomy and it already has many large volumes of medical ontologies for different but particular purposes. Gene ontology (GO) for classification of medical terminology, G2D for disease to gene, Hugo for human gene nomenclature, OMIM (Online Mendelian Inheritance in Man): a catalog of human genes and genetic disorders, GDB (Gene database), Ensembl, and LocusLink are major instances. Those biomedical search engines based on ontologies have unique ids of their own indexing but are related with diseases and relevant genes and proteins. GUI part of OnSSA takes into account of the relation of ontologies. Since one of the strengths of the Semantic Web is the distribution of the available information among a lot of nodes. Our distributed search systems assume the existence of several processing agents and each system provides a particular way of identifying systematic search for literature reviews for a decision-making on behalf of consumers, policymakers and clinicians. Periodically, agents review their success and report general success and selected results to the consulting agent through the mining agent. The reliability of distributed information agent is

determined depending on how close and/or related the search results of individual information agents is to the query. The evaluation of search results retrieved by individual information agent is done by the mining agent and handed to the consulting agent acting as a supervisor. With the evaluation result, this supervisor enhances the whole state of the best agent with the selected results, exchanges agents that are not performing well, and then communicates the enhanced state as new start state to all agents while interacting with the user. The generalization and/or specialization of query are based on the query model that represents systematic search strategy at the level of user interface. The query models are enhanced with new knowledge that it has learned from the analysis of results returned. In this paper, we mainly focus on how a secondary query is formulated and how differently information agents search a secondary query reformulated from a first query by a user. The details of the rest modules are not addressed in this paper.

The system supports the best match ranked output retrieval with a query. DAG(direct acyclic graph)-based query models provide plausible queries based on a query by the user. For example, if a novice user inputs a disease name, either the system can regenerate the query with relevant genes and get a confirmation with the user or the user can choose a button to reformulate the query with the relevant genes. The adapted query is then distributed to multiple information agents capable of operating the query but with a different search strategy through its own ontology. The query model here represents *what* the user wants to do, while the agent ontology network is *how* the information is retrieved.

Construction of a Secondary Query using Query Model

Our system differs from other user interface systems in that the system reformulates a query given by a user autonomously and proactively to a more adequate and relevant query to search in a massive and professional domain. Major components of an agent can be following: taxonomy of classes is represented in RDF schema, properties of a class can be modeled as a schema to represent intelligent states of an agent. The state itself is represented into RDF graphs and the graphs consist of both facts that agents should know in advance and knowledge obtained through an agent's perception. In addition, RuleML[8][9] plays major roles with following properties: First, it allows to define integrity constraints of avoiding illegal intelligent state of an agent. Second, it can describe knowledge of agent properties or that of agent's learning process through derived rules. Third, it can define reaction rules for an agent to respond to events and/or messages.

Fig. 3 demonstrates a simplified diagram of Query Model to show the process how a query given by a user is reformulated through Jena[11], Jess[12] and SweetJess[13]. UserQuery is accepted in a RDF form where (s p o) are subject, predicate and object respectively.

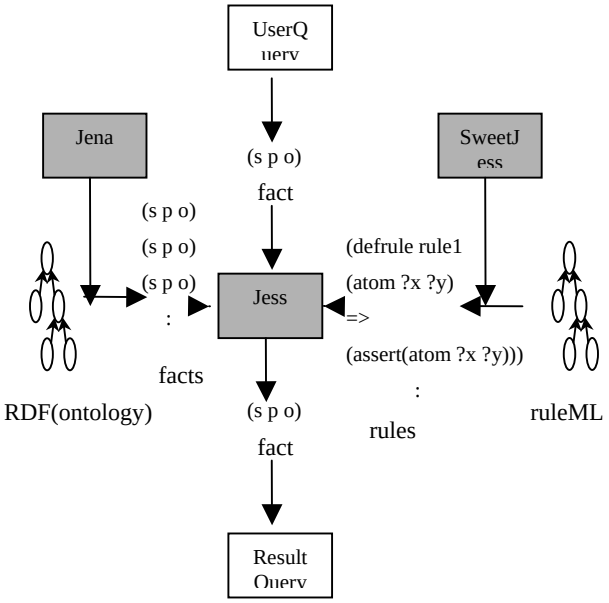

Fig. 3. Simplified Diagram of QueryModel

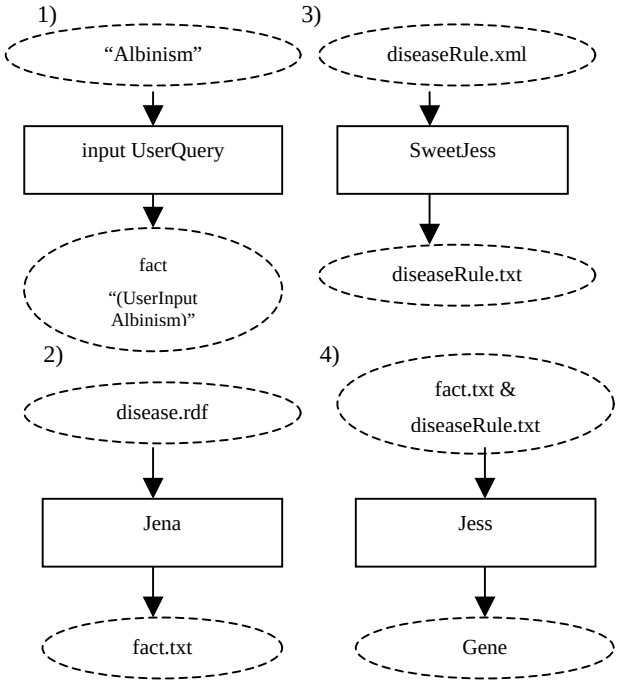

Fig. 4. Example of User Query ="Albinism"

The detailed process of reformulating a secondary query is represented with an example of user query with "Albinism" in Fig. 4. Each step of the process : 1), 2), 3), and 4) in Fig. 4 is represented with matching Input/Output in Fig. 5.

1)

```
-input UserQuery-
String inputQuery=args[0];
String userfact="(UserInput "+args[0]+")";
```

2)
```
-input-
<rdf:RDF
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:disease='http://idis.catholic.ac.kr/disease#'
 >
 <rdf:Description rdf:about='Disease'>
  <disease:DN>Albinism</disease:DN>
  <disease:SDN rdf:resource='#Albinism'/>
 </rdf:Description>
 <rdf:Description rdf:about='#Albinism'>
  <disease:DN>Oculocutaneous</disease:DN>
  <disease:DN>Ocular</disease:DN>
  <disease:DN>Partial</disease:DN>
-Jena-

String rdfFile="disease.rdf";
InputStream in=readingRDF.class
        .getClassLoader()
        .getResourceAsStream(rdfFile);
model.read(new InputStreamReader(in), "");
FileOutputStream fos= new FileOutputStream("facts.txt");
OutputStreamWriter osw = new OutputStreamWriter(fos);
BufferedWriter bw = new BufferedWriter(osw);
```

```
-output-
<Disease> <http://idis.catholic.ac.kr/disease#DN> "Albinism"
<#Albinism> <http://idis.catholic.ac.kr/disease#DN> "Ocular"
<Disease> <http://idis.catholic.ac.kr/disease#SDN> <#Albinism>
<#Albinism> <http://idis.catholic.ac.kr/disease#DN> "Partial"
<GeneDisease> <http://idis.catholic.ac.kr/disease#DN> "Albinism"
<#Albinism> <http://idis.catholic.ac.kr/disease#DN>
"Oculocutaneous"
```

3)
```
-input-
<?xml version="1.0" encoding="UTF-8"?>
<rulebase xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://userpages.umbc.edu/~m
gandh1/2002/06/RuleML/ruleml-sclp-prag-v13.xsd"
direction="forward">
  <imp>
   <_rlab> <ind>rule1</ind> </_rlab>….
<_opr> <rel>UserInput</rel></_opr>….
  </imp>
</rulebase>
```

```
-SweetJess-
Sweetjess sweetjess=new Sweetjess ();
sweetjess.transRJ("diseaseRule.xml", "diseaseRule.txt");
```

```
-output-
(defrule rule1  (GeneDisease ?type ?query)(UserInput ?query)
=> (assert (Result ?query gene)))
```

4)
```
-input-
(defquery search
(declare (variables ?x))
(Result ?x ?y))
(deffacts data
(Disease http://idiscatholicackr/disease#DN Albinism)
(#Albinism http://idiscatholicackr/disease#DN Ocular)
(Disease http://idiscatholicackr/disease#SDN #Albinism)
(#Albinism http://idiscatholicackr/disease#DN Partial)
(GeneDisease http://idiscatholicackr/disease#DN Albinism)
(GeneDisease http://idiscatholicackr/disease#DN cough)
(#Albinism http://idiscatholicackr/disease#DN Oculocutaneous)
(inputQuery Albinism)
)
(defrule r1
(GeneDisease ?type ?query)
(inputQuery ?query)
=>
(assert (Result ?query gene))
```

```
-Jess-

Rete r=new Rete();
r.executeCommand("(defquery search (declare (variables ?x))
(Result ?x ?y))");
r.executeCommand("(deffacts data"+facts+")");
r.executeCommand(rules);
r.executeCommand("(run)");
r.store("RESULT", r.runQuery("search",
        new ValueVector().add(new Value(inputQuery, RU.ATOM))));
r.executeCommand("(store RESULT (run-query search "+inputQuery+"))");
```

```
-Jess Processing-

(bind ?it (run-query search Albinism))

(while (?it hasNext)
(bind ?token (call ?it next))
(bind ?fact (call ?token fact 1))
(bind ?slot (fact-slot-value ?fact __data))
```

```
-output-
(Result Albinism gene)
```

Fig. 5 Input/Output of the Example Process

Relation of Hugo, GDB, OMIM databases

Within OnSSA different medical ontologies such as Hugo, GDB, OMIM, LocusLink are utilized to retrieve documents related to a query in addition to its own ontology. Hugo lets the agent retrieve approved human gene names related to a disease name in the query. Each gene symbol has links to other databases. Each database provides references of each gene with its own ids and these references are correlated through gene names. GDB provides the genes' GDB ID with scores of indicating genes' relevance to the disease. Here the information agent could take different search strategies by using either OMIM references or correlation information of different references for each gene provided by LocusLink. In our experiments, we examined both approaches in which both of them used GDB scores for re-

ordering gene names. GDB has Hugo's gene symbols as a primary name and provides three possible accession and each gene has a score ranking data. The information agent generates PubMed ids matching with the genes through OMIM id. It formulates an adaptive query with both a disease name and the PubMed ids of relevant genes and submits the query to PubMed for literature retrieval. Fig. 6 (a) shows an agent ontology of this information agent in a ontology network and Fig. 6 (b) presents the agent ontology into TRIPLE/DAML+OIL language[10]. The relations among Hugo, GDB, and OMIM are represented in DAML+OIL.

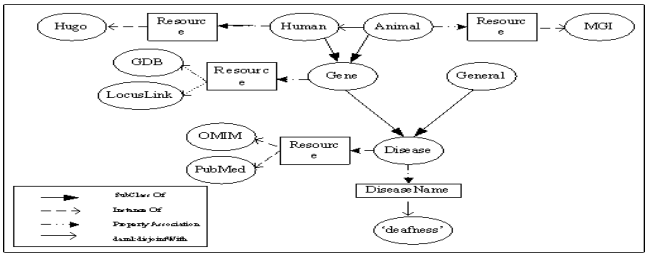

Fig. 6. (a) system ontology network

```
daml := 'http://www.daml.org/.../daml+ oil#'.
localAgent := 'http://localhost/localAgent#'.
@ localAgent:ontology {
 localAgent:Disease[rdf:type -> daml:Class].
 localAgent:Gene[rdf:type -> daml:Class; rdfs:subClassOf -> localAgent:Disease].
 localAgent:General[rdf:type -> daml:Class; rdfs:subClassOf -> localAgent:Disease;
    daml:disjointWith -> localAgent:Gene].
localAgent:Human[rdf:type -> daml:Class; rdfs:subClassOf -> localAgent:Gene].
localAgent:Animal[rdf:type -> daml:Class; rdfs:subClassOf -> localAgent:Gene;
daml:disjointWith -> localAgent:Human].
FORALL Mdl @rdfschema(Mdl){ //model block
            FORALL O,P,V O[P->V] <- O[P->V] @Mdl. // copy triples from Mdl

...FORALL O,P,V O[subClassOf -> V] <-
            EXISTS W (O[subClassOf -> W] AND W[subClassOf -> V]).
}
```

Fig. 6. (b) TRIPLE/DAML+OIL

Evaluation of Information Agent Performance

Each group of retrieved literatures by an information agent is evaluated for relevancy to a query given. This indicates the utility of utilizing the ontology within a query model for a secondary query. The relevancy level is a real number indicating how close the literatures are to the query. Each query term is denoted as *t* and its associated relevancy level as *L(t)*. The evaluation is done by computing *L(t)*, after each information agent returns a group of documents, by : $L(t) = (0.5 + 0.5\ freq_{i,q} / max_l\ freq_{l,q}) * \log n/m$ with *n* as the number of relevant documents containing this term *t* and *m* as the number of relevant documents. *L(t)* of a simple query only through PubMed is used as a baseline of others for comparison. Depending on the expertise of the user on this domain, the user can provide feedback directly to the returned results. Another role of the agent system is the learning process. It can distinguish the user interests with subject matters from retrieved information and enhance the user profile and/or agent ontology network. The validation of agent ontology is also needed as it reinforces its knowledge. The details of empirical result and evaluation can be found in the paper [14].

FUTURE WORK

Other more advanced search engines are based on self-learning principles. But how well these systems learn and how they learn correctly is important so therefore, we are examining self-learning algorithm such as Bayesian network for agent ontology. Further we will attempt to include user profiles in order to help users by providing correlated information through user's individual interests and/or genetic inheritances.

ACKNOWLEDGMENTS

We thank SungJoon Joo and DukWhan Park for their valuable help for implementation and this work was supported by grant No.(R05-2002-000-01351-0) from Korea Science & Engineering Foundation.

REFERENCES

[1] J. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations. New York:PWS Publishing Co., 1999

[2] M. Uschold and R. Jasper, A Framework for Understanding and Classifying Ontology Applications, In Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods.

[3] P. Karp, An Ontology for biological function based on molecular interactions, Bioinformatics 16(3):269-285, 2000

[4] Semantic Web in W3C http://www.w3c.org/2001/sw

[5] D. Hunt and K. McKibbon, Locating and appraising systematic reviews., Ann Intern Med. 1997;126:532-8

[6] J. Glanville and C. Lefebvre, Identifying systematic reviews: key resources ACP J Club. 2000; 132:A11-2

[7] L. Bero and A. Jadad, How consumers and policymakers can use systematic reviews for decision making, Ann Intern Med. 1997; 127:37-42

[8] A. Eberhart, An Agent Infrastructure based on Semantic Web Standards, submitted to Elsevier Science, May 2002

[9] H. Boley, S. Tabet, G. Wagner, Design rationale of RuleML: A markup language for Semantic Web rules, in Semantic Web Working Symposium, 2001.

[10] TRIPLE Semantic Web, http://triple.semanticweb.org/, March, 2002

[11] Jena, http://www.hpl.hp.com/semweb/doc/tutorial/DAML/

[12] JESS: Java Expert System Shell, http://herzberg.ca.sandia.gov/jess/docs/52/api/jess/Rete.html

[13] B. Grosof, M. Gandhe, and T. Finin, SweetJess: Translating DameRuleML to Jess

[14] J.J Yang, An Ontology-based Intelligent Agent System for Semantic Search in Medicine, to be appeared in PRIMA03, November, 2003.