A SEMANTIC SEARCH ENGINE FOR SPATIAL WEB PORTALS

Wenwen Li, Chaowei Yang

Joint Center for Intelligent Spatial Computing, George Mason University
Fairfax, VA, 22030
{wli6, cyang3}@gmu.edu

ABSTRACT

Spatial Web Portals have dramatically improved the sharing, exchanging and interoperating of Earth Science data, information and services. Currently, huge amount of geospatial data, metadata, and web services have been collected, cataloged, and made available through SWPs to serve a broad user community. However, most search engines in SWPs are based on keyword matching, which can not effectively 'understand' the meaning of users' queries, especially when a user has limited Earth science knowledge. So how to find needed data from a variety of geospatial resources becomes a big challenge. In this paper, we'll discuss a way of building semantic search engine to make the spatial-aware search more intelligent.

Index Terms— semantic, ontology, reasoning

1. INTRODUCTION

Spatial Web Portal (SWP) [1] utilizes the advantages of web portal techniques, such as easy to configure, share and integrate, is widely used by the Earth science community in Earth science data sharing and exchanging. Popular SWPs include NASA's Earth Science Gateway (ESG), ESIP's Earth Information Exchange (EIE, http://eie.cos.gmu.edu), FGDC's Geospatial One Stop (GOS, http://gos2.geodata.gov) Portal, NASA's Earth Observing System Clearinghouse (ECHO, http://www.echo.nasa.gov/), NASA's Global Change Master Directory (GCMD, http://gcmd.nasa.gov/), and NOAA's National Climatic Data Center (NCDC, http://www.ncdc.noaa.gov). These SWPs store a large amount of geospatial resources, including text files, raw and post-processed data, and various geospatial web services. However, the popular utilizations also stand out problems of how to find needed data from a variety of geospatial resources and how to visualize the data from multi-perspectives. First, the search tools within the SWP are based on full-text match technique, such as Lucene, which cannot find terms with similar meanings and different appearance. This can be categorized as synonym problem [2]. The second drawback is that putting the same term into different context will have different meanings, for example, "Washington" could be a person, a place or food. But current search engines in the SWPs cannot distinguish the differences. This is categorized as polyseme problem [2]. To solve the above problems, we try to utilize semantic web techniques [3] to enhance traditional search by building a domain knowledge base (KB) and do semantic reasoning from the KB. Section 2 discusses the important elements of a semantic search engine. Section 3 describes the system architecture. Section 4 demonstrates the prototype. At last, section 5 concludes and discusses future research goals.

2. IMPORTANT ELEMENTS

There are several essential elements to compose a spatial-aware semantic search engine: (1) an information model used to build the KB for a specific area and an inference service which can parse the semantic meaning of user's query based on the model; (2) one or more data sources as back-end repository; (3) furthermore, a mechanism to extract and visualize spatial web services automatically from the result webpages. In section 2.1-2.4, we'll discuss how we address these issues.

2.1. Ontology Based Information Model

Our model is based on SWEET (Semantic Web for Earth and Environmental Terminology) ontology, where all the terminologies are defined in different facets, including phenomena, property, substance, earth realm [4]. Terminologies in one facet are organized into tree hierarchy with simple relationship of inheritance, such as Parent-Child and Siblings. And terms from different facets would be connected with specific relationships to form an integrated network. As the most popular ontology model in Earth Science, SWEET provides an upper-level abstracted expression of the world. Based on formal description logic (DL), it can support TBox reasoning, which reasons assertions on concepts, such as a set of classes and properties. But using this model alone may not be enough on reasoning ABox (assertions on individuals) since no individuals are defined in the ontology. The individuals in ABox are instances of classes, for example, “Hurricane” is a terminology defined in SWEET ontology, it has several properties including “maximum sustained wind speed”, “location”, “duration” and etc. And hurricanes “Katrina” and “Isabel” are instances of the class “Hurricane”. To realize both TBox and ABox reasoning, we extended SWEET ontology and added relative individual information. Figure 1 shows an ontology fragment in air quality domain. In this figure, circles are classes, blocks are instances and arrows which connect classes/classes or classes/instances are properties. Classes with different colors are mapped to different facets including phenomena, substance, and property of SWEET ontology. Instances in red are the part of ontology we extended from SWEET. In this way, the KB with expressive logic correlations is well built for semantic reasoning.

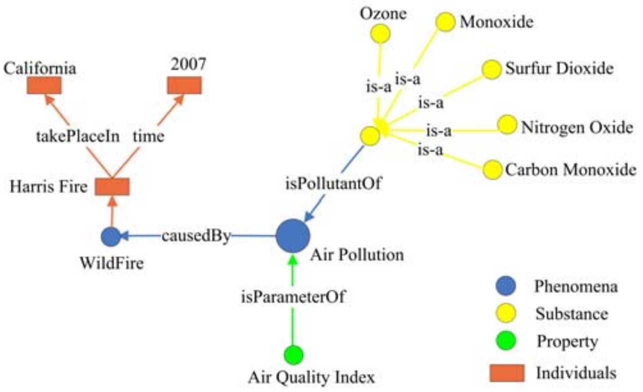


Figure 1. A graph of ontology fragment pertaining air pollution. The semantics expressed by the ontology are: a) Substance like “Ozone”, “Monoxide”, “Sulfur Dioxide”, “Nitrogen Oxide”, “Carbon Monoxide” and etc are pollutant of “Air Pollution”. b) Property “Air Quality Index” is a parameter of phenomena “Air Pollution”; c) Phenomena “Air Pollution” is caused by Phenomena “Wildfire”; d) there is an instance of wildfire called “Harris Fire” taking place in California in 2007.

2.2. Semantic Reasoning

Semantic reasoning is the core part of the semantic search engine. Given a user’s query, syntax analysis, semantic analysis and retrieval tasks from heterogeneous environment are performed in sequence. Syntax analysis focuses on analyzing components of a query sentence, as well as getting the ‘central word’, i.e. the exact object that user is interested in. This would help us to retrieve the KB and get proper candidates efficiently. It can be done by either providing a query template for user to map phrases into different dimensions, such as “WHAT” or “HOW”, provided in GUI or relying on a natural language parser, such as Stanford’s open source statistical parser [5]. In this paper, we use the first method to save client parsing time as well as to get more “accurate” information from user’s query. After syntax analysis, user’s query is mapped into two levels: logic level and formal query level for semantic reasoning. When reasoning is conducted, complex queries will be decomposed to sub-queries. For example, after California fire, people may ask “What’s the air pollution caused by California fire in 2007?” Through syntax analysis, we get that “air pollution” can be distinguished as event, which is also the central word of the whole sentence, “fire” as reason, “California” as place, “2007” as time. Given these information, the natural language descript query can be transformed to a DL-based query:

Q1:AirPollution$\cap\exists$causeBy.Fire$\cap\forall$takePlacein.California$\forall$ hasTime.2007

Emanating from the central word “Air Pollution” of Q1, we could retrieve more useful information based on knowledge stored in the KB, which is the process of query decomposition. For example, referring to the ontology fragment in Figure 1, Q1 could be decomposed as:

Q1a: Parameter$\cap\exists$isParameterOf.AirPollution
Q1b: Fire. Name$\cap\exists$cause.AirPollution$\cap\forall$takePlacein.CA$\forall$ Happenin.2007
Q1c: Pollutant$\cap\forall$isPollutantOf.(AirPollution$\cap\forall$causedBy. Fire)

Where, Q1a aims to find the names<n_1, n_2…n_k> of all the fires that satisfy the restrictions, it is an ABox statement and need reasoning. Q1b and Q1c are formed by checking all the roles that are connected with ‘Air Pollution’ to get the pollutants<Po_1, Po_2…, Po_m> and parameter<Pa> that’s used for measuring air pollution separately. Both of them are TBox statements which need TBox reasoning.

The inference engine we used for TBox and ABox reasoning is Jena Semantic Web Framework for Java [6] Reasoning procedures are: (1) Ontology is loaded into memory or persistent storage maintained by Jena; (2) the DL-based query are transformed into formal SPARQL query [7] (3) Though the query APIs that Jena provide, the sub queries are conducted and results for different sub queries are retrieved. (4) Query results are then combined in an appropriate manner to get expanded and more specific information. In addition, we also realized implicit association inference by traversing the ontology tree that the query term belongs to recursively. This is important because the associations of a class should not only contain its own associations but also its ascendants’ associations.

2.3. Data Sources and Geo-Bridge

With the expanded information we get from inference service, the system will redirect these information to multiple SWPs through geo-bridge. It works under two modes: if the spatial portals provide web APIs to search from their databases, such as NASA's GCMD portal and NOAA's NCDC portal, Geo-bridge will redirect user's query through the APIs, and then extract URL links, titles, abstracts from the returned records and send them back our client. For portals which provide CSW (Web Catalog Service) [8] interfaces, such as FGDC's GOS portal and NASA's ESG portal, Geo-Bridge will post XML-encapsulated request to the interface. For other portal which provides its own API interface, such as NASA's ECHO portal, Geo-bridge will customize the codes to build the connection. Then by parsing the returned XML document, it'll be able to present the results to end users. In addition, Geo-bridge introduces AJAX technique in client and remote portal interaction, this asynchronous communication mechanism makes data exchange efficiently.

2.4. Auto Web Map Service Extraction

Web Map Service (WMS) [9] is a very popular service to enable interoperability. It provides an intuitive way for spatial data exchange and visualization. Thus automatically detecting and extracting WMS information from result webpage will make it more convenient for users to view the data. Furthermore, whether to contain a WMS service in a dataset description is also an important criterion for ranking the results. In general, three locations in the result page are important to be considered as containing a WMS service: first, the hyperlinks with hypertext containing "WMS" or similiar information. Sometimes, the judgment of hyperlink is difficult because the hyperlink itself doesn't link to a WMS service, but links to another page which can redirect to a WMS service after the link is clicked, this situation occurs in GCMD portal. To get WMS information from these complicated hyperlinks, we need to do pre-analysis and customized work. The second location might be anywhere on the webpage, this is always important because sometimes WMS service providers will put the service URL on the webpage but not in the format of a hyperlink. A way to detect WMS service like this is to utilize regular expression analyzing method to detect the strings which starts with "http://" but not as a hyperlink, and then further analysis, such as send "GetCapabilities" or "GetMap" request could confirm whether it is a WMS service. The third location exists in the XML encapsulated metadata file of a dataset. There will be a field of "OnlineSource" or "Servicelink", which provides the WMS service link. The fieldname should be different according to different metadata schemas. If any service is found, the service will be linked to both a 2D and 3D map client by a pop-up button for visualization and other operations.

3. ARCHITECTURE

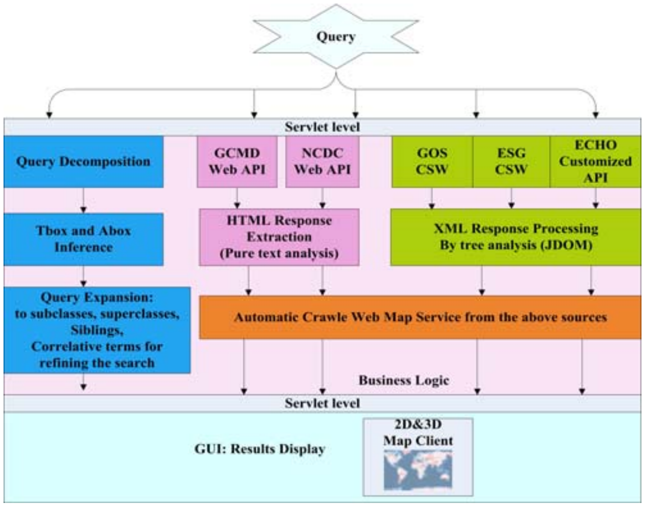
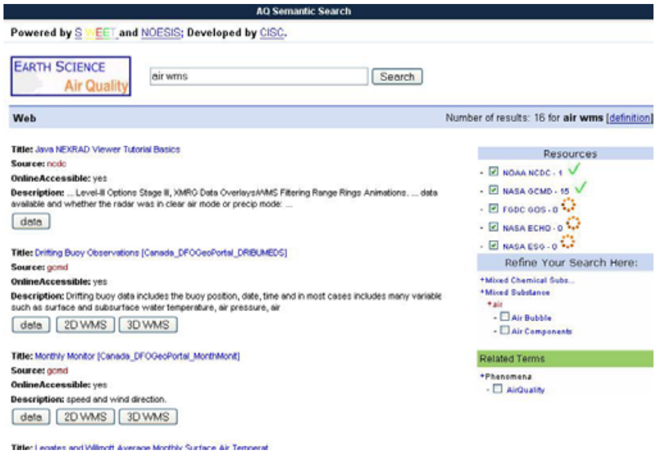

Query
Servlet level
Query Decomposition
GCMD Web API
NCDC Web API
GOS CSW
ESG CSW
ECHO Customized API
Tbox and Abox Inference
HTML Response Extraction (Pure text analysis)
XML Response Processing By tree analysis (JDOM)
Query Expansion: to subclasses, superclasses, Siblings, Correlative terms for refining the search
Automatic Crawle Web Map Service from the above sources
Business Logic
Servlet level
GUI: Results Display
2D&3D Map Client

Figure 2. System Architecture

Figure 2 shows the system architecture of our semantic search engine. It could be divided into several layers: graphic user interface layer or client deals with user's query request and communicates with the servlet layer. Servlet layer is a middle layer which is used to redirect query keywords to business logic layer as well as send results back to the client. The servlet layers showed at the top and bottom of the architecture are the same, and query and GUI is also the same layer. We depart them to show the workflow clearly. Business logic layer communicates with multiple data resources and does semantic reasoning. This layer is the most important layer in the architecture and the work mode has been introduced in the above section. Here we mainly introduce the data interfaces and communication modes between each two layers. From the GUI layer, user's query keywords will be structured as a query array and send to servlet layer through AJAX. The data returned from GCMD, NCDC, GOS, ESG and ECHO are stored in a class. When the data are returned to servlet layer, the fields are extracted and encapsulated into a XML document through a uniform schema. When the data are ready at the server side, AJAX in client side gets response, and retrieves the XML document, then structures to a tree display style. In addition, we also implemented dynamic writing mechanism by starting threads when accessing different data resources in the server side. In the client side, a timer is set to access the server to get the latest data. Once the query is done in the server side, the thread is killed and a flag is sent to client side to stop the timer. This mechanism helps to reduce the response dramatically.

4. PROTOTYPE

Figure 3 demonstrates the prototype of semantic search engine. Left side of the GUI is searching result with notation of where the web documents come from; the upper right side of the GUI designates data sources, with a progress bar indicating the process when the searching is conducted to each resources; the lower right side of the GUI provides refine search suggestion (back from inference service) with search keyword's parents, children, siblings and associated terms.



Number of results: 16 for **air wms** [definition]

Title: Java NEXRAD Viewer Tutorial Basics
Source: ncdc
OnlineAccessible: yes
Description: ... Level-II Options Stage II, XMRG Data OverlaysWMS Filtering Range Rings Animations. ... data available and whether the radar was in clear air mode or precip mode: ...

Title: Drifting Buoy Observations [Canada_DFOGeoPortal_DRIBUMEDS]
Source: gcmd
OnlineAccessible: yes
Description: Drifting buoy data includes the buoy position, date, time and in most cases includes many variable such as surface and subsurface water temperature, air pressure, air

Title: Monthly Monitor [Canada_DFOGeoPortal_MonthMonit]
Source: gcmd
OnlineAccessible: yes
Description: speed and wind direction.

Title: Legates and Willmott Average Monthly Surface Air Temperat

Figure 3 Semantic Search Engine for Air Quality Community in EIE portal

5. CONCLUSIONS AND DISCUSSION

This paper discusses the spatial-aware search problem in current SWPs, and based on that, it proposes a method to construct a semantic search engine to improve the search efficiency. The method has several advantages: First, it provides users a uniform platform to search, view and operate spatial information. Second, the main components especially those for ontology and semantic reasoning are coupled loosely, which would be easy to inherit ontologies of other domain into the semantic search engines. The only issue is we need to modify some connections to portals in other domains. In the future, we aim to record user's search behavior, including their preferences, their search habits and etc. This would help ontology engineers to improve our KB continuously. Moreover, we would expand the search engine to support other areas, such as Water, Carbon, Agriculture, Coastal, Public Health, Invasive Species, and Disaster.

6. REFERENCES

[1] C. Yang, J. Evans, M. Cole, N. Alameh, S. Marley, and M. Bambacus, "The Emerging Concepts and Applications of the Spatial Web Portal" , *PE&RS* , 73(6):691-698, 2006.

[2] M. Nauman, S. Khan, M. Amin, and F. Hussain, "Resolving Lexical Ambiguities in Folksonomy Based Search Systems through Common Sense and Personalization", Proceedings of the Workshop on Semantic Search at the 5th European Semantic Web Conference, Tenerife, Spain, pp 2-13, 2008.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", *Scientific American* 284, 34–43, 2001.

[4] R. Raskin and M. Pan, "Semantic Web for Earth and Environmental Terminology (SWEET)", *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, Florida, USA, 2003.

[5] D. Klein and C. D. Manning, "Fast Exact Inference with a Factored Model for Natural Language Parsing", *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge, MA, MIT Press, pp. 3-10, 2003.

[6] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne and K. Wilkinson, "Jena: Implementing the semantic web recommendations", *Proc. of the 13th Int. World Wide Web Conference (WWW 2004)*, 2004.

[7] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF", http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/, 2005.

[8] Nebert, D., Whiteside, A., Vretanos, P. (Eds.). Catalogue Services Implementation Specification, Version 2.0.2, OGC Document Number: 07-006r1, Open Geospatial Consortium, U.S., 218pp, 2007.

[9] Beaujardiere, J. (Eds.). Web Map Service Implementation Specifications, Version 1.3, OGC Document Number: 04-024, Open Geospatial Consortium, U.S., 85pp, 2004.