Towards Natural Language Interfaces for Geospatial Analysis

Upendra Dadi[1,2], Liping Di[3]

[1]Earth Resources Technology, Inc, Laurel, MD [2]National Oceanographic Data Center, Silver Spring, MD [3]Center for Spatial Information Science and Systems, Fairfax, VA

Introduction

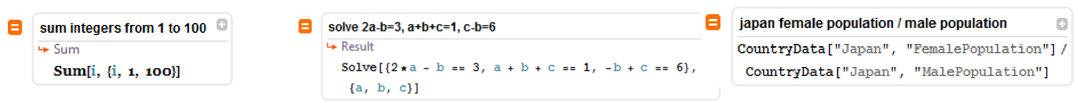
Complex data analysis tasks require software tools which are sophisticated. Currently, such tools take huge amount of time and resources to learn and use. Even if one is domain expert with good understanding of algorithms and techniques implemented by the software, one would still need to learn the idiosyncrasies of the software environment being used. Users of the software have to constantly refer to the documentation to perform tasks which are not routine. Ideally, the software should be easy enough to learn and use at the level of an expert in a short span of time, especially by those who already have fair amount of knowledge of the theory, algorithms and methods used by the software. At the extreme, the user doesn't have to have only minimal knowledge about the software tool, she/he would state the task to be performed in natural language using the terminology from the domain of study and the software would understand the query and respond appropriately.

Goal

The goal of this project is to create a software tool which responds to simple natural language geospatial queries over geospatial basemaps used in a given domain. The tool will be supplied with ontologies relevant to answering the queries . The ontologies would consist of ontologies of location related to the domain of study. Initially, the tool would respond to simple geometric, topological, raster and attribute queries posed in natural language over the base map data. The tool should to able to interpret a series of natural language queries in a dalogue and respond appropriately. When an ambiguous query is posed, it should respond with a set of ranked alternatives which are probable matches to the given query.

Related Work

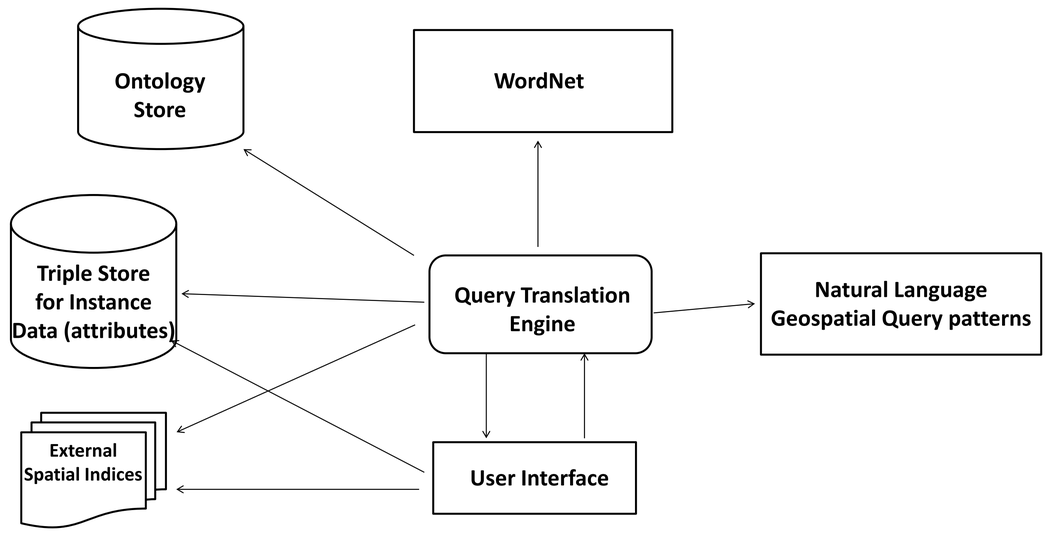
Natural Language Interfaces have been developed for several analysis software. Mathematica 8 uses free-form linguistics[1] which allows users to enter queries in plain English. The users don't have to remember the commands traditionally used for performing the tasks. The natural language text is automatically converted to Mathematica syntax. The interface greatly improves the usability of the software. Here is a basic example of using Mathematica[2] using free-form linguistics:



sEnligh (system English) could be used for interfacing with MATLAB. sEnglish uses a subset of English with meaning of sentences pre-defined in a high level programming language. "A correctly formulated sEnglish text compiles into program code unambiguously if predefined sentence structures and ontology is defined."[3]. Here is an example of sEnglish code which is used with a controller of an autonomous rover vehicle[3]:
"Find your current position Pc. Define Hd as a 'heading direction'. Execute " Hd = Pnxt-Pc; ". Detect obstacle position Obst in heading direction Hd. If Obst is empty, then move with heading direction Hd. If Obst is not empty, then do the following. Compute turned heading direction Hds from Hd. Detect obstacle position Obst2 in heading direction Hds. If Obst2 is empty, then move in heading direction Hds. If Obst2 is not empty, then do the following. Compute turned heading direction Hds2 from Hds. Detect obstacle position Obst3 in heading direction Hds2. If Obst3 is empty, then move in heading direction Hds2. Finish conditional actions for second heading. Finish conditional actions for first heading."

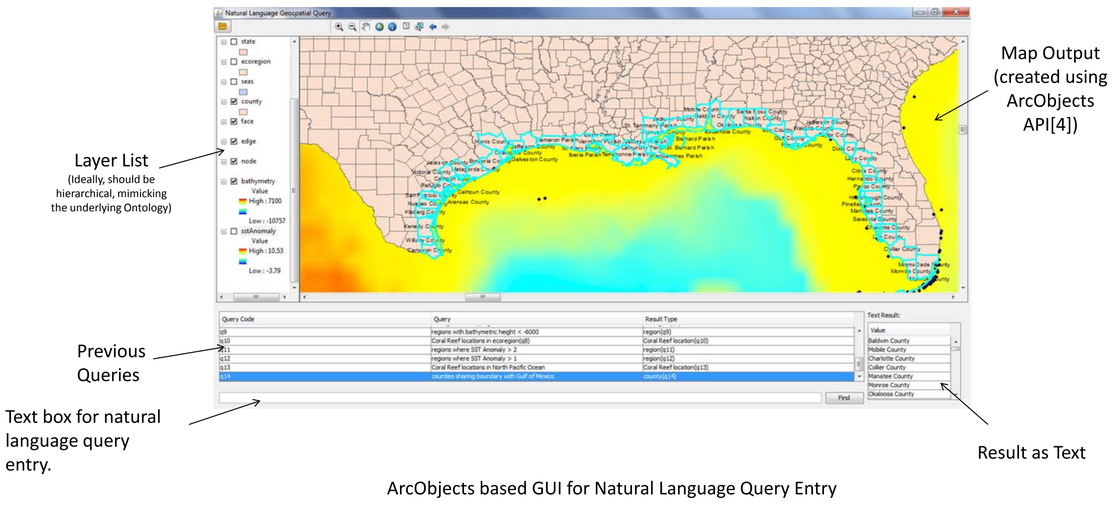
This paragraph is translated to a high level programming code when supplied by suitable knowledge base of an ontology and sentence definitions.

System Architecture



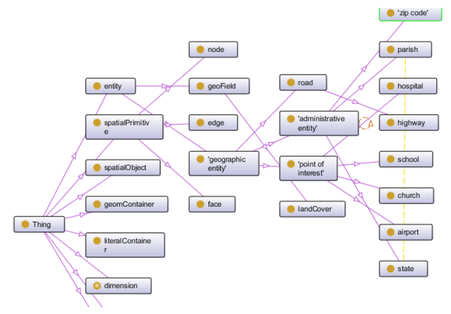
The figure above shows the system architecture. Central to this architecture is the Query Translation Engine which translates the Natural Language Query to SPARQL query or other high level code. The query translation engine communicates with Ontology store and instance store to identify all the keywords in the user input. On finding duplicate keywords, it responds back to the user for disambiguation. The engine follows two stage process to correctly interpret the sentence: 1) It looks into the query pattern database to see if there is a syntax match between the user input and a query pattern. Each matched query has the appropriate translation. 2) If no match is found, it uses WordNet[4] to find a pattern which is closest. The robustness of the system can be improved either by increasing the number of patterns in the database and in the limit have every pattern possible or by using sophisticated sentence matching algorithms which try to match one sentence with another based on the meaning of the sentence.

Query Interface



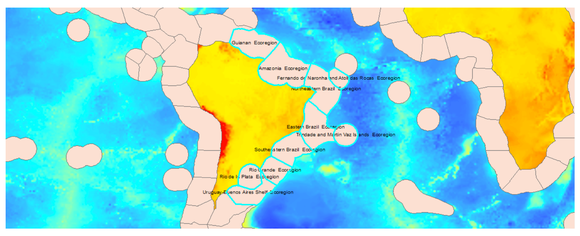
ArcObjects based GUI for Natural Language Query Entry

Several ontologies were developed to store the relationship between various geographic features within a given domain. The picture below shows an ontology based on Geographic Names Information System feature classes.

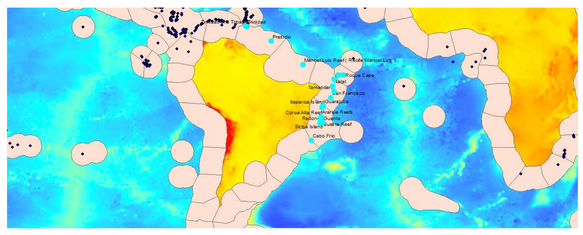


An ontology based on the GNIS feature classes

Some Example Queries

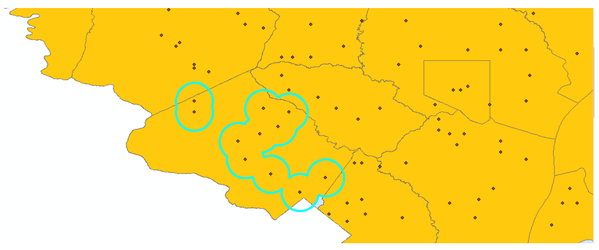


Query: "Marine ecoregions overlapping with Brazil"

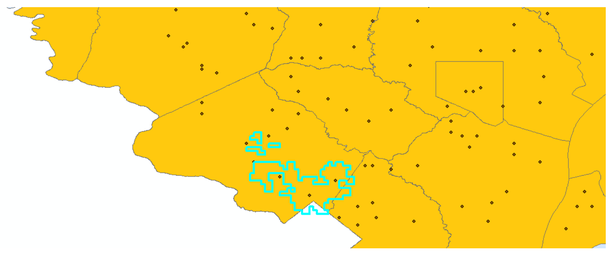


Query(contd. from previous query): "Coral Reef locations within the marine ecoregions"

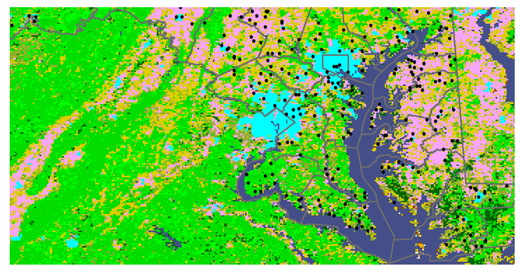
Query: "Airports in Montgomery County"



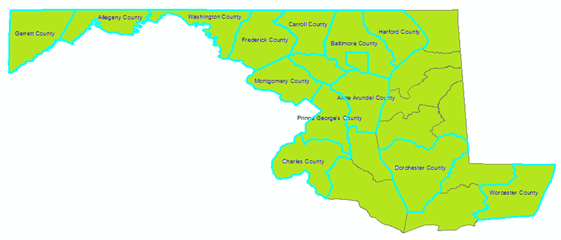
Query(contd.): "Region within 100 miles of the airports"



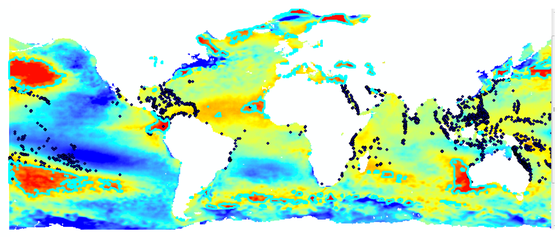
Query(contd.): "Areas in the region where AVHRR land cover code is equal to 14"



AVHRR land cover grid[6]



Query: "Land area of counties in Maryland. Mean of the land area. Counties with land area greater than the mean."



Query: "Regions where SST Anomaly > 1"

Discussion

A true natural language query interface would allow user to enter any valid sentence. But it is found that without some constraints on the user input, the system becomes too complex to manage. Translating a large sentence with several conjunctions and disjunctions into code is extremely complex. But it was found that most of the complex sentences can be broken down to a set of simpler sentences which could be handled much more easily. Currently, the system requires the user to enter simple sentences which can be glued together to answer a particular query. To give an example, the query "all the counties in Maryland whose land area is greater than mean land area of counties in Maryland." can be translated to "land area of counties in Maryland. Mean of the land areas. Counties with land area greater than the mean." It might be possible to do rudimentary natural language programming by using variables in the sentence. For example, the above query can be seen as an instance of a more generic query which is applicable over all states, not just Maryland. One can define "larger counties" of a state as counties who area is larger than the mean. The definition can now be used repeatedly to perform queries.

There are several benefits to storing feature data in a triple store than using tables. For example, traditional GIS do not handle a hierarchy of features efficiently. The triple store can handle such relations more efficiently. Also, triple stores are natural fit for storing topological data which are graph based. But one still would have to depend on external spatial indices for geometric queries.

Conclusion

The use of natural language querying interfaces for geospatial analysis has some fundamental advantages over traditional menu driven or command line based interfaces. Certain operations are still performed best by the click of mouse, like zooming and panning on a map, than a verbose sentence. But for many other operations, having a natural language interface would help greatly.

References

The datasets used in this poster are from the following sources:

- VLIZ Marine Gazetter. http://www.vliz.be/vmdcdata/vlimar/about.php
- U.S. Census Bureau Tiger Database. www.census.gov/geo/www/tiger/
- U.S. Geographic Name Information System. http://geonames.usgs.gov/pls/gnispublic/
- NOAA Operational SST Anomaly Charts. http://www.osdpd.noaa.gov/ml/ocean/sst/anomaly.html

1) Mathematica Free-Form Linguistics: http://www.wolfram.com/mathematica/new-in-8/free-form-linguistic-input/
2) Mathematica Free-Form Linguistics. Basic Examples: http://www.wolfram.com/mathematica/new-in-8/free-form-linguistic-input/basic-examples.html
3) sEnglish: http://www.system-english.com/
4) WordNet: A Lexical Database for English, Miller, GA, Communications of the ACM, Vol 38 Issue 11, Nov. 1995
5) ArcObjects: http://help.arcgis.com/en/sdk/10.0/arcobjects_net/ao_home.html
6) University of Maryland, AVHRR land cover, Global Land Cover Facility: http://www.glcf.umd.edu/data/landcover/